



**Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Desplegament anàlisi i millora d'un sistema distribuït de mesures en xarxes de nova generació.

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Víctor Sáez García

DIRECTORS: David Rincón Rivera
Frederic Raspall Chaure

DATA: 22 de Novembre de 2010

Títol: Desplegament, anàlisi i millora d'un sistema distribuït de mesures en xarxes de nova generació.

Autor: Víctor Sáez García

Directors: David Rincón Rivera
Frederic Raspall Chaure

Data: 22 de Novembre de 2010

Resum

A l'actualitat, les xarxes són un element essencial en àmbits com la recerca i l'educació, així que per a un bon funcionament d'aquest canal de comunicació cal desenvolupar eines que permetin el control de les comunicacions extrem a extrem.

El projecte Géant2 connecta prop de trenta milions d'usuaris (estudiants, professors, investigadors) en una topologia multi-domini que s'estén per trenta-quatre països al continent europeu. Per a la monitorització de les transmissions entre diferents xarxes nacionals de recerca, s'han desenvolupat les eines del conjunt perfSONAR.

Aquests serveis permeten la monitorització d'aquest tipus d'entorn on apareixen diferents operadors, permetent als usuaris d'una xarxa A verificar el correcte funcionament de les transmissions cap a una xarxa B, sense que l'usuari tingui permisos d'administració a la segona xarxa.

A la primera part d'aquest treball es comenten els serveis de la distribució MDM de perfSONAR, els components de cadascun dels serveis i la seva instal·lació. S'ha aconseguit la instal·lació de tots els components de la distribució i s'han configurat i realitzat proves amb els serveis 'Lookup Service', 'RRD-MA', 'BWCTL'.

La segona part del document analitza la interfície gràfica que es proporciona, la perfSONAR UI i els seus components. També es facilita una guia per a la generació de noves funcionalitats per l'eina i una sèrie d'aplicacions d'exemple.

Title: Deployment, analysis and improvement of a distributed measurement system in next generation networks

Author: Víctor Sáez García

Directors: David Rincón Rivera
Frederic Raspall Chaure

Date: November, 22th 2010

Overview

Computer networks are an essential element in research and education, and therefore, in order to obtain a good performance of this communication channel, there is a need to develop tools that allow controlling this communications end to end.

The Géant2 project connects over thirty million users (students, professors, researchers) with a multi-domain topology that expands across thirty-four European countries. perfSONAR has been developed in order to monitor the transmissions all over the associated national research and education networks.

These services allow monitoring in this kind of environments with several network operators, making possible for a user in the network A to verify the transmission to a network B without having administrator rights on the second network

In the first part of this document the perfSONAR MDM services and their components are introduced and installed. The installation of all services has been achieved, and 'Lookup Service', 'RRD-MA' and 'BWCTL-MP' services have been successfully installed and tested.

The second part of the thesis contains an analysis of the user interface (perfSONAR UI) and its components. A guide for deploying new tools for the interface is also provided, as well as some example tools.

**Voldria donar les
gràcies als meus pares,
Ana i Sisku per tot el
suport rebut, la
paciència que han
tingut amb mi, i per
extensió, a tota la
família.**

**També a tots els
companys i amics que
m'han acompanyat
durant aquest temps. I
un record especial per
la gent de Cornellà.**

**Gràcies també al David
Rincón i al Frederic
Raspall per la seva
col·laboració i
implicació durant la
realització d'aquest
treball.**

ÍNDEX

INTRODUCCIÓ	1
CAPÍTOL 1. ENTORN PERFSONAR	3
1.1. Què es perfSONAR	3
1.1.1. Entitats creadores de perfSONAR	3
1.1.2. Que ens permet perfSONAR.....	7
1.1.3. Distribucions de perfSONAR	8
1.1.4. Serveis de perfSONAR.....	9
1.1.5. Distribució escollida	12
CAPÍTOL 2. PREPARACIÓ DEL SISTEMA I SERVEIS BÀSICS.....	13
2.1. Material	13
2.2. Prerequisits	14
2.2.1. Definició dels repositoris de perfSONAR	14
2.2.2. Màquina virtual de Java.....	14
2.2.3. Apache Tomcat	14
2.2.4. eXist.....	15
2.3. Lookup Service.....	16
2.3.1. Prerequisits.....	16
2.3.2. Funcionament del servei	16
2.3.3. Els missatges de registre a serveis de <i>Lookup</i>	18
2.3.4. Problemes durant la instal·lació del servei.....	18
2.4. Authentication Service.....	18
2.4.1. Sistema de validació.....	19
2.4.2. Prerequisits.....	20
2.4.3. Recull de problemes durant la instal·lació del servei	20
CAPITOL 3. ELS SERVEIS D'ARXIU DE MESURES.....	21
3.1 RRD-MA	21
3.1.1 <i>Round Robin Database</i>	22
3.1.2 Prerequisits del servei	23
3.1.3 Recull de problemes durant la instal·lació del servei	24
3.1.4 L'arxiu de configuració.....	24
3.1.5 Obtenció de dades	25
3.1.6 Mètode alternatiu de publicació de dades.....	26
3.1.7 Publicació de dades	27
3.1.8 Revisió de permisos	28
3.2 EI SQL-MA	29
3.2.1 Prerequisits.....	29
3.2.2 Preparació del servei MySQL.....	30
3.2.3 Obtenció de dades	30
3.2.4 Publicació de dades	31
3.2.5 Arxiu de configuració del servei.....	32
3.2.6 Passarel·la Ibatis	33
3.2.7 Dificultats trobades durant la instal·lació del servei	33

CAPITOL 4. ELS PUNTS DE MESURA	34
4.1. EI BWCTL-MP	34
4.1.1 Prerequisits	35
4.1.2 Problemes durant la instal·lació del servei	36
4.1.3 Verificació del funcionament del servei	37
4.1.4 Proves realitzades amb el servei BWCTL-MP	38
4.2. EI Telnet/SSH	38
4.2.1. Prerequisits	39
4.2.2. Comprovació del servei	39
4.2.3. Recull de problemes durant la instal·lació del servei	39
CAPITOL 5. LA INTERFÍCIE GRÀFICA DE PERFSONAR	40
5.1 Instal·lació de l'eina	40
5.2 Funcionalitats de l'eina	41
5.2.1 Interfaces	41
5.2.2 Circuits	42
5.2.3 BWCTL	43
5.2.4 Looking Glass	43
5.2.5 FlowSA	44
5.2.6 Playground	45
5.2.7 Lookup Service Playground	45
5.3 Desenvolupament de noves funcionalitats	46
5.3.1 Preparació de l'entorn de desenvolupament	46
5.3.2 Afegint noves aplicacions	46
CAPITOL 6. ALTRES ACTIVITATS, LÍNIES FUTURES I CONCLUSIONS....	51
6.1 Altres tasques realitzades	51
6.2 Estudi d'ambientalització	51
6.3 Línies futures	51
6.4 Conclusions	52
BIBLIOGRAFIA	54
GLOSSARI D'ABREVIATURES UTILITZADES EN AQUEST DOCUMENT ..	57
ANNEX A. ARXIUS DE CONFIGURACIÓ DEL SERVEI APTITUDE.....	59
ANNEX B. INSTAL·LACIÓ I CONFIGURACIÓ DELS PREREQUISITS	60
B.1 Definició dels prerequisits de perfSONAR	60
B.2 Java Virtual Machine	60
B.3 Apache Tomcat	61
ANNEX C. EL PANELL DE CONTROL WEB D'EXIST	65
C.1 El panell web d'eXist	65

C.2	Funcionament del motor de bases de dades eXist	67
ANNEX D. INSTAL·LACIÓ I CONFIGURACIÓ DEL LOOKUP SERVICE 71		
D.1	Instal·lació del servei	71
D.2	Comprovació del servei	71
D.3	Configuració del servei	74
D.4	Configuració del servei de <i>lookup</i> local per la resta de serveis	75
ANNEX E. LLISTA DE SERVIDORS GLOBALS DE PERFSONAR 77		
ANNEX F. INSTAL·LACIÓ I CONFIGURACIÓ DEL AUTHENTICATION SERVICE..... 78		
F.1	Instal·lació del servei	78
F.2	Verificació de la instal·lació	78
F.3	Problemes trobats durant la instal·lació del servei.....	78
ANNEX G. INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI RRD-MA 83		
G.1	Problema amb la instal·lació de la llibreria librrd2.....	83
G.2	Instal·lació del servei	85
G.3	Comprovació del servei.....	85
G.4	Configuració del servei	85
G.5	Generació de l'arxiu de configuració del servei.....	87
G.6	Creació de les bases de dades RRD	88
G.7	Verificació de les mesures.....	91
G.8	Script col·lector de dades	95
ANNEX H. CODI PER A LA PUBLICACIO DE MESURES A TRAVES DEL SERVEI RRD-MA..... 97		
ANNEX I. INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI SQL-MA..... 99		
I.1	Instal·lació.....	99
I.2	Configuració del servei	99
ANNEX J. CODI PER A LA PUBLICACIÓ DE MESURES D'UTILITZACIÓ AL SERVEI SQL-MA 100		
ANNEX K.INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI BWCTL-MP..... 103		
K.1	Prerequisits.....	103
K.2	Instal·lació del servei	106
K.3	Proves amb el servei BWCTL	107
ANNEX L.INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI TELNET/SSH 114		
L.1	Instal·lació.....	114
L.2	Configuració bàsica del servei.....	114
L.3	Configuració del servei	115
ANNEX M. ARXIUS DE CONFIGURACIO DE SERVEIS A LA INTERFÍCIE GRÀFICA..... 117		
ANNEX N. IMPLEMENTACIÓ MINIMA DE LES CLASSES PERFSONARTAB I MAINCOMPONENT 118		

ANNEX O. CODI DE LA PESTANYA MOSTRATEXT	122
ANNEX P. CODI DE LA PESTANYA PINGERSQL	130
ANNEX Q. CODI DE LA PESTANYA SQL RETRIEVER	137
ANNEX R. CODI PER A LA PESTANYA SOCKET CONNECTOR	141
ANNEX S. ARXIUS PROPERTIES SERVEI NDT	148

INTRODUCCIÓ

El perfSONAR és una eina que permet l'emmagatzematge i l'intercanvi d'informació de funcionament de xarxes en un entorn multi-domini federat. El seu objectiu és ser el punt de consulta d'aquesta informació per a simplificar l'administració de grans xarxes, facilitar la resolució d'incidències entre dominis i fins i tot, permetre a les aplicacions de nova generació condicionar el seu funcionament depenent de l'estat de la xarxa.

Aquest projecte neix de la mà de l'associació de tres grans xarxes de recerca, Géant2, Internet2 i RNP així com d'altres *partners* interessats com ara ESNET.

El sistema perfSONAR vol anar més enllà de les eines de monitorització habituals consistents en un mecanisme de proves i un punt d'emmagatzematge, permetent a operadors d'altres xarxes de la federació, la consulta de les dades dels serveis locals.

En aquests moments existeixen diferents distribucions de perfSONAR. Totes comparteixen les funcionalitats i són compatibles entre sí, permetent als administradors locals escollir la distribució que més s'adapti al seu entorn

Aquest projecte té com a primer objectiu analitzar, documentar i provar la instal·lació i el funcionament del conjunt d'eines proporcionades per perfSONAR, així com aquelles eines individuals de les quals es compona. Com a segona finalitat, s'analitza, documenta i es proporcionen una sèrie de noves funcionalitats d'exemple per a la interfície gràfica perfSONAR UI.

L'objectiu d'aquestes tasques és servir de principi per al desenvolupament de noves funcionalitats per a les eines de perfSONAR dintre del projecte Géant3, una associació de xarxes nacionals de recerca i educació.

Així doncs, a la primera part del treball s'analitzen, s'instal·len i es documenten les eines que conformen la distribució perfSONAR MDM 3.2. S'han instal·lat correctament totes les eines, i s'han configurat i realitzat proves amb alguns dels serveis indicats.

La segona part del projecte presenta les funcionalitats que incorpora la interfície gràfica perfSONAR UI i analitza el funcionament a nivell de codi. També es presenten determinades aplicacions que poden servir d'exemple per a futurs desenvolupaments.

El capítol 1 fa una introducció a l'entorn perfSONAR, presentant les entitats que promouen la seva utilització, per a continuar comentant els serveis que proporciona i les diferents distribucions disponibles. També es comenta la distribució escollida per a la realització d'aquest treball.

El capítol 2 comença comentant el material que es disposa per a la realització del treball, i després s'introdueixen els prerequisits que té qualsevol dels

serveis de perfSONAR, que són el servidor Apache Tomcat, el motor de base de dades eXist i la màquina virtual de Java. El capítol continua presentant els primers serveis de perfSONAR, que són aquells que donen un valor afegit com a sistema de monitorització multi-domini, el servei d'anunci o *Lookup Service* i el servei d'Autenticació.

El següent capítol, el número 3, es centra en els serveis d'arxiu de mesures, és a dir els serveis que emmagatzemen l'històric del funcionament dels elements de xarxa que s'hi defineixen. perfSONAR proporciona dos serveis d'aquest tipus on la major diferència resideix en els motors de bases de dades utilitzats, tipus *Round Robin* i *SQL*. També s'analitzen mètodes de captació i publicació de dades.

El capítol 4 correspon als serveis de punt de mesures, és a dir els serveis que ens permetran realitzar proves contra altres xarxes. En aquest apartat es presenten els serveis de mesures *BWCTL* i *Telnet/SSH*. Ambdós han estat instal·lats correctament. I de la mateixa manera que en els altres capítols, també s'analitzen algunes de les eines que componen els serveis.

El darrer capítol pràctic és el 5è que està dedicat plenament a la perfSONAR UI, el punt d'accés als serveis per els usuaris del sistema. A la primera part del capítol es mostren les opcions que permet la interfície actualment i a la segona part es donen indicacions per al desenvolupament de noves eines. Es presenten algunes peces de codi desenvolupades per a servir d'exemple.

El darrer capítol d'aquest document comenta altres tasques que realitzades que tenen relació amb l'entorn de perfSONAR, les futures línies de treball on aquest document pot fer de referència, les conclusions de les tasques realitzades, i inclou també unes línies sobre l'estudi d'ambientalització.

CAPÍTOL 1. ENTORN PERFSONAR

Aquest capítol comença explicant el perquè de l'aparició d'un sistema amb les característiques de perfSONAR, per després continuar comentant les diferents distribucions existents de perfSONAR i acabar presentant els diferents serveis de perfSONAR de forma individual, que en el seu conjunt formen les distribucions.

1.1. Què es perfSONAR

perfSONAR [1], o *PERformance Service Oriented Network monitoring ARchitecture*, és un software de mesura i monitorització de xarxes separades físicament, però connectades entre si en un entorn federat, que permet la consulta i realització de proves en els serveis desplegats a tot el domini.

També es pot entendre com diferents conceptes:

- Un consorci d'organitzacions amb l'ànim de crear eines d'anàlisi de xarxes operable entre diferents dominis, útil per a l'anàlisi de la pròpia xarxa i per a l'anàlisi de la comunicació amb d'altres, sent un dels principals objectius facilitar la resolució de problemes extrem a extrem.
- Un protocol on s'estableixen una sèrie de rols, es defineix una manera de comunicar-se entre ells estàndard per tal de que els serveis es comuniquin entre sí.
- Un conjunt de codi que permeti l'anàlisi de xarxes operable entre xarxes diferents, sent algunes peces d'aquest codi més important, per ser el que permet aquesta operabilitat entre dominis.

1.1.1. Entitats creadores de perfSONAR

El perfSONAR entès com a un consorci d'organitzacions, apareix de la mà de les següents:

1.1.1.1. Géant2

El projecte Géant2 [2], consisteix en la creació, d'una xarxa d'alta velocitat dedicada a la comunitat de recerca i educació que connecta prop de trenta milions d'usuaris a Europa entre estudiants, professors i investigadors..

És una xarxa multi-domini que s'estén per trenta-quatre països europeus i enllaça amb d'altres regions del món.

Géant2 és un projecte co-fundat entre la Unió Europea i les xarxes nacionals de recerca i educació, gestionada per DANTE. DANTE, que correspon a '*Delivery of Advanced Network Technology to Europe*', s'estableix a Cambridge

La figura 1.1 mostra el mapa de la topologia de la xarxa Géant2 prevista per a finals de 2010, on hi apareixen els principals enllaços de comunicació entre les diferents xarxes nacionals.

De cara al projecte Géant3, en fase de desenvolupament, s'hi afegixen les xarxes de recerca de Macedònia, Sèrbia i Montenegro. A més el projecte contempla que la majoria dels enllaços entre les xarxes nacionals de recerca tinguin una capacitat de 10Gbps o superiors, arribant als 40 i 100 Gbps. També es vol permetre l'establiment d'enllaços dedicats de manera dinàmica i sota demanda que podrien superar els 10Gbps utilitzant tècniques WDM (*Wavelet Dimension Multiplexing*).

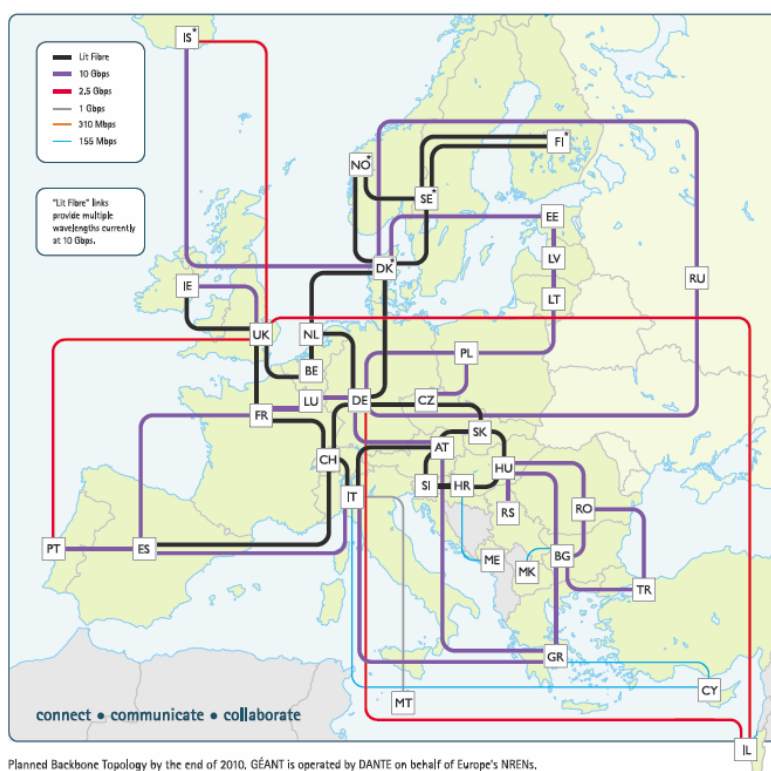


Fig. 1.1 Mapa de la topologia de la xarxa Géant2 prevista per final de 2010

Un dels membres d'aquesta xarxa és RedIRIS [3], la xarxa nacional de centres de recerca i educació (en anglès *National Research and Education Networks* o *NREN*) d'Espanya, adscrita a Géant. El seu nom prové de Red Española para Interconexión de los Recursos Informáticos. Es va fundar a l'any 1988 com a part d'un projecte de R+D per part del Ministerio de Educación y Ciencia amb la col·laboració de Telefónica. A l'actualitat RedIRIS es gestiona des de l'entitat pública empresarial Red.es i és finançada pel Pla Estatal de R+D+i. La figura 1.2 mostra la topologia actual de la xarxa RedIRIS.

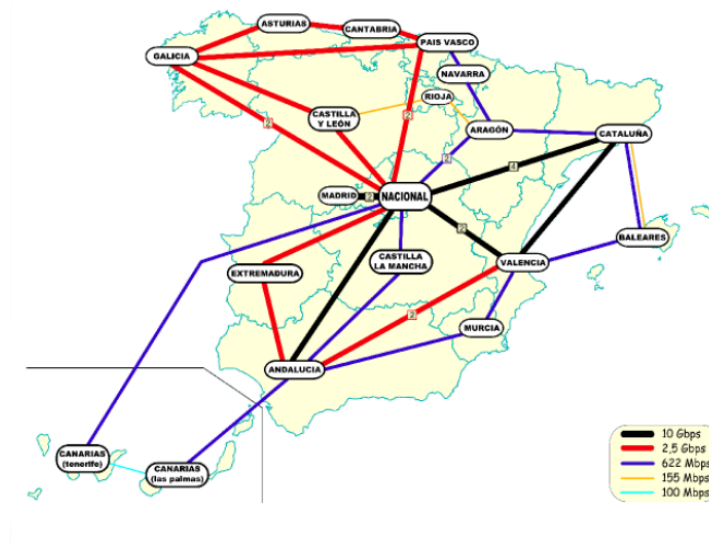


Fig 1.2 Desplegament de la xarxa RedIRIS

1.1.1.2. Internet2

El projecte Internet2 [3] es pot considerar com l'equivalent nord-americà de Géant2. Establert al 1996, és una organització sense ànim de lucre formada per més de dues-centes universitats americanes en cooperació amb setanta grans empreses, quaranta-cinc agències governamentals, laboratoris i d'altres institucions. La seva missió també consisteix en el disseny i implementació de xarxes d'alt rendiment per a la comunitat científica i de recerca, animant també als seus integrants a desenvolupar noves eines que permetin millores als sistemes actuals i per tant contribuir a la millora global d'Internet.

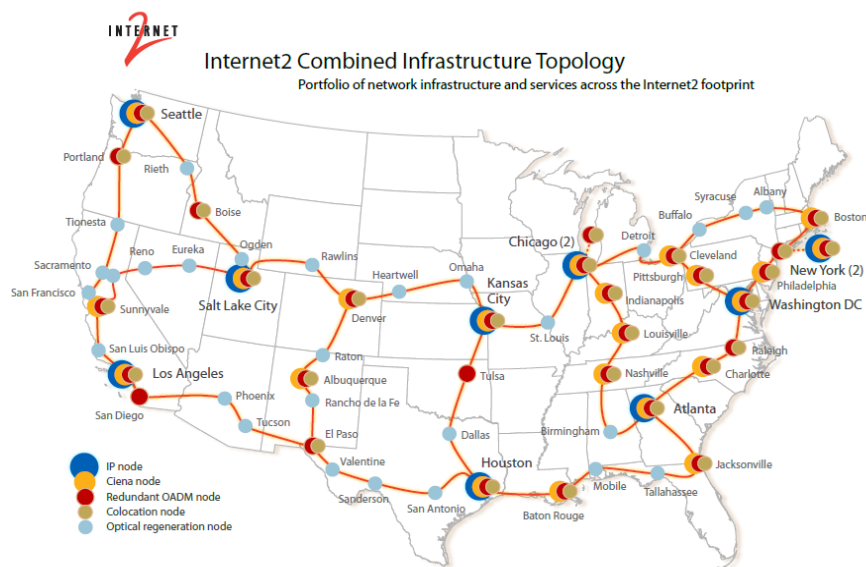


Fig 1.3 Mapa dels Estats Units d'Amèrica amb els enllaços de la xarxa d'educació i recerca d'Internet2

1.1.1.3. Esnet

L'*Energy Sciences Network* [4], és un consorci també dedicat a la provisió d'enllaços amb gran ample de banda per a investigadors, universitats i altres institucions, permetent a aquests la seva col·laboració en importants projectes científics relacionats amb l'energia i la climatologia entre d'altres recerques científiques. Situat a Berkeley, al *Lawrence Berkeley National Laboratory*, en el moment de l'elaboració d'aquest treball està en marxa el projecte ESnet4, consistent en una xarxa de 10 Gbps que connecta les institucions adherides.

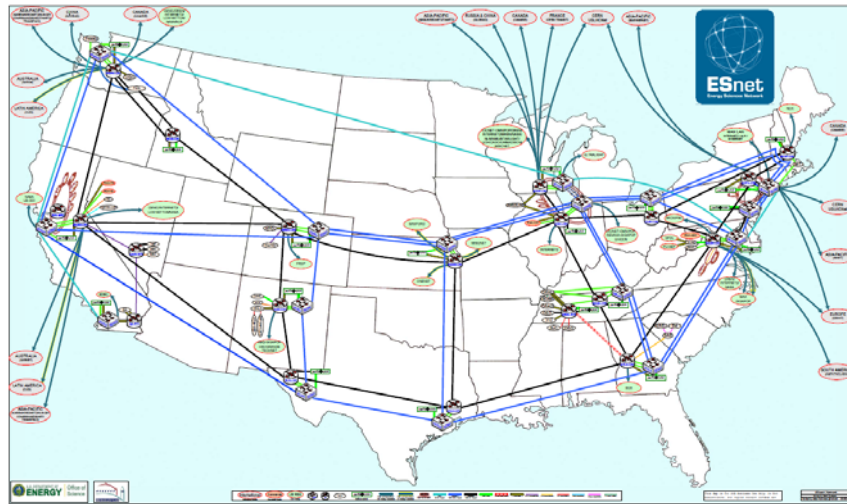


Fig 1.4 Topologia de la xarxa de recerca d'Energy Sciences Network

1.1.1.4. RNP

La *Rede Nacional de Ensino e Pesquisa*, es alhora l'equivalent brasiler de la xarxa Rediris a Espanya. És l'encarregada de la connexió entre els centres de recerca i universitats al Brasil. A l'actualitat es troben desplegats en vint-i-set punts, un a cadascun dels estats del Brasil i un altre al districte federal, connectant quinze xarxes estatals i més de sis-centes institucions.

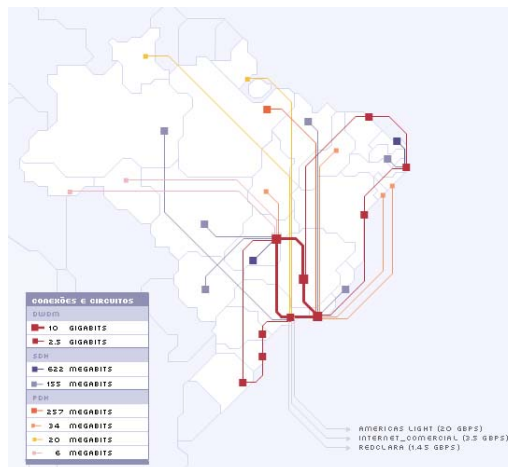


Fig 1.5 Mapa del Brasil amb els enllaços que conformen la xarxa RNP

1.1.2. Què ens permet perfSONAR

El perfSONAR és un conjunt de mòduls que funcionen de forma individual, pensats per a emmagatzemar i monitoritzar dades de funcionament de xarxes entre dominis diferents que formen part d'un global. El sistema s'està desenvolupant per a utilitzar-se com a eina per l'administració de grans xarxes, facilitar la resolució de problemes de comunicació extrem a extrem, o condicionar el comportament d'aplicacions segons l'estat d'una xarxa.

El perfSONAR consta de diferents tipus de serveis que es poden classificar de la següent manera:

- Servei d'anunci de serveis (o de *lookup*)
- Servei d'autenticació
- Arxius de mesura
- Punts de mesura

Aquests serveis es descriuen en més detall a l'apartat 1.1.4.

El fet que perfSONAR pretengui ser una eina útil per a la resolució de problemes de comunicació extrem a extrem implica un problema amb les fronteres físiques i organitzatives de cadascuna de les subxarxes que formen la xarxa global. Analitzem el cas de Géant, com a exemple. El recorregut entre dos centres de recerca pot passar per diferents xarxes. Per exemple, la xarxa de l'EETAC està connectada a la xarxa de la Universitat Politècnica de Catalunya, que alhora està connectada a la xarxa Rediris, la xarxa d'universitats a nivell nacional que a la seva vegada connecta a la xarxa de Geant. Per tant, s'han de buscar mètodes que permetin tenir una situació global, d'un entorn multi-domini a tots els usuaris que ho requereixin.

Suposem que dos investigadors, que col·laboren en un mateix projecte, vulguin realitzar una transferència de dades entre la nostra escola i el centre de supercomputació de Poznan, seguint un camí com el que es mostra a la figura 1.6.

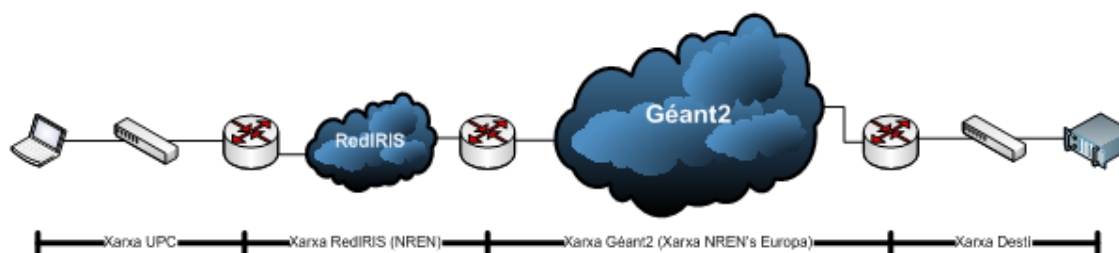


Fig 1.6 Esquema del camí d'una transmissió entre centres adscrits a la xarxa Géant2.

L'investigador de la UPC se n'adona que la transmissió no s'està fent correctament i avisa al servei tècnic del seu centre. El tècnic farà ús de les

eines corresponents per a determinar si hi ha algun problema a la xarxa local, a la qual ell té els permisos adients per a realitzar les accions que calgués.

Si el tècnic diagnostica que el problema es produeix fora de la xarxa de l'escola, en condicions normals, l'administrador no podria determinar on es localitza el problema. Aquí és on entra en joc el sistema de perfSONAR, ja que realitzant les proves pertinents, el sistema perfSONAR permetrà al tècnic veure l'estat de les comunicacions entre la UPC i la resta de nodes adscrits a perfSONAR, podent valorar en quin punt es produeix la fallida i tot i no tenir permisos d'administració fora de la xarxa de l'escola, entre la xarxa RedIRIS i la xarxa destí com es mostrava a la figura 1.6.

Així doncs, apareixen dues condicions que el sistema perfSONAR ha de complir, la disponibilitat i la seguretat que les proporcionen el servei d'anunci i el servei d'autenticació respectivament.

Per començar, el sistema haurà de saber de tots els nodes disponibles que implementin aquesta tecnologia per a que l'usuari pugui escollir en cada moment el servei al que vol accedir. Això s'aconsegueix mitjançant el servei d'anunci o de lookup. El sistema perfSONAR té dedicat un servidor central que s'encarrega de registrar aquells servidors perfSONAR 'locals' que s'adhereixin al servei global.

Un sistema d'aquestes característiques permetria, a priori, que un usuari qualsevol accedís a dades de funcionament de xarxes a les quals no n'ha de tenir accés, de manera que l'altra condició que ha d'acomplir el sistema és l'autenticació, que permetrà l'accés en cas de tenir unes credencials correctes.

Els serveis d'arxiu de mesures, com el seu nom indica, consisteixen en dos serveis que permeten l'accés a les dades de funcionament que hi hagin emmagatzemades als diferents servidors. A través de la interfície proporcionada per perfSONAR, es pot consultar les característiques i les dades de funcionament d'enllaços que s'hagin definit.

perfSONAR també proporciona serveis de punts de mesura, que permeten la realització de proves al moment, amb l'objectiu que el tècnic que hagi de resoldre una incidència, pugui fer proves al instant.

1.1.3. Distribucions de perfSONAR

Un cop descrits breument els serveis de perfSONAR, comentarem les diferents maneres en les que podem descarregar les eines per a la implementació dels serveis [5].

1.1.3.1. perfSONAR MDM

Es correspon a l'eina perfSONAR *Multi-Domain Monitoring*. Apareix com a eina multidomini en la versió 3 de perfSONAR cap al Maig de 2008.

Aquesta distribució consta d'una sèrie de paquets Unix que contenen una implementació en Java dels serveis necessaris per a la utilització del sistema. Introdueix també interfícies de configuració via web, per a una administració més còmode.

1.1.3.2. perfSONAR PS

Aquesta distribució, inclou els mateixos serveis i implementa els mateixos protocols que la versió *MDM*, utilitzant per a la seva construcció codi *Perl*. Està totalment pensada per a ser compatible amb qualsevol altre sistema que implementi els protocols de perfSONAR.

1.1.3.3. perfSONAR Performance Toolkit

A diferència de les anteriors distribucions, el perfSONAR *Performance Toolkit* és una distribució *Knoppix* customitzada per a ser utilitzar-se des de un *Live CD*. En aquest cas, el software consisteix en una imatge ISO disponible a la plana del servei.

Aquesta imatge es pot fer servir directament gravant-la en un CD i arrancant el equip des de el CD o bé llançant la imatge ISO amb una màquina virtual. Un cop arrencat el sistema, aquest ens demanarà una sèrie de paràmetres de configuració. Un cop configurats, disposarem d'un sistema totalment preparat per utilitzar els serveis de perfSONAR.

1.1.3.4. perfSONAR UI

A part d'aquestes aplicacions que implementen els serveis anteriorment mencionats, l'entorn perfSONAR requereix també d'una eina per a la consulta per part d'usuaris. Aquesta eina és la perfSONAR *User Interface*, la interfície gràfica d'usuari. Ens permetrà doncs fer consultes als serveis que hi hagi disponibles a aquells servidors perfSONAR accessibles o que s'hi defineixin.

1.1.4. Serveis de perfSONAR

Ara que coneixem els diferents tipus de perfSONAR disponibles, és el moment de donar més detalls sobre els diferents serveis i les seves funcionalitats.

1.1.4.1. Lookup Service

El *Lookup Service (LS)* és el servei encarregat d'indicar al usuari quins son els serveis disponibles a la màquina que està consultant. Aquesta eina és una de les encarregades de donar una visibilitat tant local com global dels serveis perfSONAR que hi ha instal·lats i funcionant arreu.

Cada cop que un servei s'arrenca, es registrarà amb el Lookup service per a indicar que està disponible i proporcionar una descripció de les seves capacitats. Aquests serveis envien missatges XML sobre HTTP al servei LS que emmagatzema en una base de dades els serveis que té disponibles i contesta indicant que el registre s'ha efectuat correctament amb un altre missatge XML.

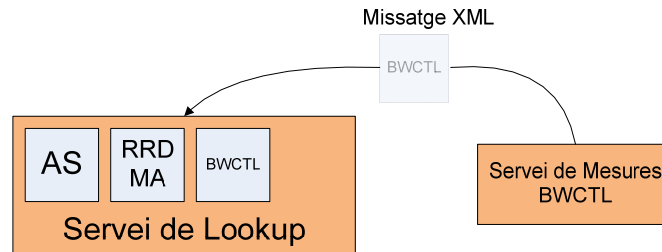


Fig. 1.7 Esquema del procés de registre d'un servei de mesures a un servei de *lookup*

A més d'aquesta funcionalitat a nivell local, el *Lookup Service* s'encarrega també de registrar-se amb d'altres serveis LS que funcionen a nivell global. Aquests *lookup services* globals (gLS) son mantinguts per alguna de les organitzacions anteriorment citades (com per exemple Géant) i cada cert temps sincronitzen amb d'altres serveis d'anunci globals per a saber en tot moment quins serveis hi ha disponibles. Els servidors locals també s'actualitzen amb aquesta informació global per assegurar que un client que faci una consulta, rebí la informació de tots els serveis que hi ha actius fora del domini.

Aquest servei no és imprescindible per al funcionament d'altres dels serveis de perfSONAR, però és de vital importància per a permetre la interoperabilitat entre dominis. Per això es molt recomanable d'instal·lar-lo abans que qualsevol dels altres serveis.

1.1.4.2. Authentication Service

El servei d'autenticació (AS) proporciona, restriccions d'accés als serveis perfSONAR en funció de les credencials de l'usuari. Un cop s'indica a un servei de perfSONAR que utilitzi l'autenticació, s'ha d'indicar també quin tipus d'accions requeriran de permisos especials per a la seva execució.

Els sistemes d'autenticació accepten identitats proporcionades per entitats adscrites a *eduGAIN* (part del projecte Géant). *eduGAIN* és un projecte que pretén establir una confederació entre els proveïdors de identitat de cadascuna de les entitats participants (*NREN's*) per tal de permetre el control d'accés a xarxes i aplicacions entre les organitzacions que hi participen [6].

Un mòdul de perfSONAR, en rebre una petició de servei que requereix d'autenticació, enviarà un missatge XML amb la informació d'autenticació. El

servei AS s'encarregarà de verificar si la identificació es correcta i contestarà amb un altre missatge XML indicant si s'ha de permetre la consulta rebuda.

1.1.4.3. Els arxius de mesures

Els arxius de mesura o *measurement archives (MA)* són mòduls que permeten l'accés a diverses dades de monitorització com:

- Utilització d'un enllaç
- Capacitat d'un enllaç
- Nombre d'errors de recepció de la interfície
- Nombre de paquets descartats

Actualment hi ha dos mòduls que posen aquestes dades a disposició dels usuaris. La seva diferència és el motor de base de dades que utilitzen. perfSONAR no indica de forma explícita com es recullen aquestes dades, però si defineixen la manera en que aquestes s'han d'emmagatzemar per a una futura consulta des de la perfSONAR UI.

Els dos serveis d'arxius són el *Round Robin Database MA* i el *SQL MA*. El primer tipus d'arxiu funciona sobre una base de dades tipus *Round Robin Database (RRD)* [7], una base de dades circular amb agregació temporal que s'explicarà amb més detall a l'apartat 3.1.1. El *SQL MA* ens dona una estructura predefinida, a la que es poden desar les dades que s'obtinguin sobre un motor SQL (*Structured Query Language*) com MySQL [8]. Modificant els scripts de configuració del servei, es possible utilitzar una base de dades que s'haguessin obtingut amb altres sistemes.

Com s'indica anteriorment, a través de la perfSONAR UI, podrem accedir a les dades i representar-les en mode text i gràfics.

1.1.4.4. Els punts de mesures

Els punts de mesura permeten fer mesures in situ sobre diferents paràmetres a sistemes perfSONAR. Actualment el sistema proveeix dos punts de mesura: *BWCTL-MP* i *Telnet/SSH*.

El *BWCTL MP (Bandwidth Test Controller Measurement Point)* executa un test d'ample de banda entre dos equips. D'aquest test es pot obtenir informació del volum d'informació que pot transmetre's entre dos equips que implementin el servei podent escollir fins i tot, el protocol de transport de les dades, TCP o UDP. Quan un usuari a través de la perfSONAR UI vol executar un test, la interfície gràfica envia un missatge XML indicant quin és el camí a provar. El servidor origen, doncs, comença a realitzar les proves de connexió amb l'eina *BWCTL* i retorna el resultat al client.

L'altre servei és el *Telnet/SSH Measurement Point*, que pot obtenir la configuració i la informació de funcionament d'elements d'una xarxa remota.

També, ens permetrà calcular el retard *RTT*, es a dir, el retard d'anada i tornada entre dues xarxes, o executar una comanda Traceroute, per a analitzar el camí que seguirà un paquet des del servei d'origen al de destí.

1.1.5. Distribució escollida

Aquest projecte pretén ser punt de partida en la possible implementació de noves aplicacions per a perfSONAR. Donat que els nostres coneixements són més orientats al llenguatge Java, i per ser aquesta la distribució recomanada per Géant i RedIRIS, es considera més adient fer servir la distribució MDM que no pas les eines basades en Perl. Així doncs, en els propers capítols descriurem el procés d'instal·lació, els components que utilitza i les funcionalitats de perfSONAR MDM.

CAPÍTOL 2. PREPARACIÓ DEL SISTEMA I SERVEIS BÀSICS

Aquest segon capítol comença detallant l'equipament utilitzat durant la realització d'aquest treball, a continuació comenta alguns serveis necessaris per a la instal·lació de la distribució perfSONAR MDM. Després d'aquests prerequisits, es presenten els dos mòduls que permeten la interoperabilitat entre dominis.

2.1. Material

Per a la realització d'aquest treball s'ha disposat d'un dels equips del laboratori C4-325 de l'EETAC. En el nostre cas es tracta d'un equip model '*HP Compaq dc5700 Microtower PC*', que ha estat anomenat '*perfhost*'.

Segons es comenta a [9], els diferents mòduls a instal·lar han estat provats amb els següents sistemes operatius (i les seves derivacions, entre parèntesi):

- Red Hat Enterprise Linux 4.x i posteriors (Fedora 8 , CentOS)
- Debian 4.0 i posteriors (Ubuntu 7.10)

Per a la realització d'aquest treball es va optar per la utilització de Debian, per ser aquest un sistema operatiu GNU basat en software lliure desenvolupat per una comunitat de desenvolupadors i usuaris. Red Hat Enterprise Linux, per altra banda, és una distribució d'un sistema operatiu GNU/Linux també gratuït, però gestionat per la companyia Red Hat.

En el moment de la preparació del sistema, s'utilitza doncs la versió 5 de Debian, que apareix el més de Febrer de l'any 2009. Es preveu l'aparició de la nova versió del sistema cap a finals de l'any 2010.

Cadascuna de les versions de Debian prenen prestat el nom de personatges de la pel·lícula *Toy Story*, de manera que la versió 5 també es coneix com Lenny. A [10] es comenta que la propera versió 6.0, que serà llançada a finals de 2010, s'anomenarà Squeeze. Durant la realització d'aquest treball, en data d'agost de 2010, els '*release managers*' de Debian anuncien que el desenvolupament de la versió 6 del sistema està aturat, i que en aquest punt no es dedicaran a implementar noves funcionalitats sinó al perfeccionament del funcionament i rendiment del sistema [11].

2.2. Prerequisits

Com comentàvem anteriorment, els serveis de perfSONAR requereixen de la instal·lació prèvia d'algunes eines per al seu funcionament. Comencem definint els repositoris de perfSONAR i a continuació es detallen els diferents serveis previs.

2.2.1. Definició dels repositoris de perfSONAR

Per assegurar una correcta instal·lació dels serveis de perfSONAR, el primer pas serà indicar al nostre servidor les adreces dels repositoris per a la descarrega dels serveis de perfSONAR.

Els repositoris de perfSONAR són mantinguts pels responsables del projecte perfSONAR MDM i en ells es pot trobar les darreres versions disponibles de tots els serveis, així com instal·lables adequadament preparats per a les diferents distribucions anteriorment comentades.

Els instal·lables poden incloure algun dels prerequisits que un servei pugui necessitar, així com una pre-configuració dels mateixos. Tot i això, es trobaran diferents entrebancs, que s'aniran comentant en els apartats corresponents. Es pot trobar la definició dels repositoris del servei Aptitude a l'annex A.

2.2.2. Màquina virtual de Java

La Màquina virtual de Java [12] o *Java Virtual Machine*, és un software específic que permet la execució de codi Java compilat en un codi binari especial, o *bytecode*, generat pel compilador Java. La màquina virtual de Java és executada pel sistema operatiu, i fa de pont entre aquest codi binari i el sistema operatiu que l'està executant. Això permet que el codi Java sigui independent del hardware sobre el que s'executa.

D'aquesta manera Sun Microsystems (ara Oracle), companyia creadora i fomentadora de la tecnologia Java, dedica recursos a la creació de màquines virtuals per a molts dels sistemes operatius disponibles, de manera que un arxiu *.class* creat en un entorn Unix, pot ser perfectament interpretat a una màquina Windows, per posar un exemple.

El procés d'instal·lació de la màquina virtual de Java es troba a l'annex B.

2.2.3. Apache Tomcat

Es pot definir ràpidament l'Apache Tomcat [13] com un servidor d'aplicacions. De manera més estesa i com a característiques fonamentals podem dir que està implementat en Java i que compleix les especificacions Java Servlet i Java Server Pages de Sun Microsystems.

D'aquesta manera s'obté un servidor web amb capacitat per a processar i generar el contingut a mostrar a l'usuari a partir de codi Java. El fet de poder executar codi farà que l'Apache pugui realitzar moltes més accions més enllà de mostrar simples planes web, permetent a l'equip executar gairebé qualsevol procés.

No s'ha de confondre l'Apache Tomcat amb l'Apache Web Server, que consisteix en una implementació en llenguatge C d'un servidor web, i que no s'utilitza com un servidor d'aplicacions.

En el cas de l'entorn perfSONAR, l'execució de codi ens permetrà per exemple:

- Configuració dels serveis a través d'una interfície web. L'usuari configura els diferents serveis a partir d'un formulari web en el qual s'hi introdueixen els paràmetres configurables. Un cop s'envien les dades, el servidor Tomcat executarà el codi corresponent per a aplicar la configuració dels serveis.
- Peticions de proves de mesura. El servidor Apache Tomcat és també la porta d'entrada de les peticions de mesura que es reben des de la consola perfSONAR UI. Rep les sol·licituds i executa els programes que la duren a terme. També s'encarrega de tornar la resposta a l'usuari.
- Peticions de resultats. També es fa la passarel·la entre l'usuari i les bases de dades que hi hagi disponibles en el servidor.

De la mateixa manera que l'apartat anterior, el procés d'instal·lació de Tomcat es troba a l'annex B.

2.2.4. eXist

eXist [14] és un motor de base dades *open source* que emmagatzema dades en format XML. Els sistemes de perfSONAR es comuniquen amb missatges tipus XML, per tant no s'ha de programar un altre capa de codi que tradueixi els missatges XML al llenguatge de la base de dades i viceversa. Així que, és totalment justificat fer servir una base de dades d'aquest estil.

2.2.4.1 Instal·lació

Tal com s'indica al punt 2.2.3.1, la instal·lació del servei eXist, es realitza juntament amb el servei Tomcat, ja que perfSONAR proporciona un paquet específic pre-configurat.

2.2.4.2 Comprovació del servei

Així doncs, per a comprovar el correcte funcionament del servidor d'aplicacions, podem accedir a la següent URL, que ens mostrarà l'entrada a l'apartat d'administració d'eXist:

- <http://perfhost:8180/exist/admin.xql>

L'annex C dona més detalls del funcionament del servei de bases de dades XML.

2.3. *Lookup Service*

El servei de Lookup o d'anunci és un dels serveis primordials de perfSONAR. Com a tal no és un prerequisit per a altres serveis com els punts de mesura, però és un servei molt important, ja que és l'encarregat de donar visió global del nostre sistema a tota la resta de serveis perfSONAR distribuïts arreu.

El procés d'instal·lació i configuració d'aquest servei es pot consultar a l'annex D.

2.3.1. Prerequisits

Per a la instal·lació d'aquest servei, es requereix haver instal·lat els prerequisits que es comenten a l'apartat 2.2.

2.3.2. Funcionament del servei

El *Lookup Service* (LS), s'encarrega d'enregistrar els serveis perfSONAR que es registrin al nostre servidor per anunciar-ho a d'altres servidors de nivell superior, de manera que un usuari d'una xarxa remota, pugui conèixer els serveis perfSONAR disponibles a la nostra xarxa.

El funcionament del servei es pot veure representat a la figura 2.3.

A més de funcionar de forma local, el *Lookup Service* s'encarrega també de registrar-se amb d'altres serveis LS que funcionen a nivell global. Aquests *lookup services* globals (gLS) son mantinguts per alguna de les organitzacions anteriorment citades (per exemple Géant) i cada cert temps sincronitzen amb d'altres gLS per a saber en tot moment quins serveis són disponibles a tot l'entorn, permetent així a tots els usuaris tenir una visió de tots els serveis de perfSONAR disponibles. A l'annex E s'inclou una llista de servidors globals.

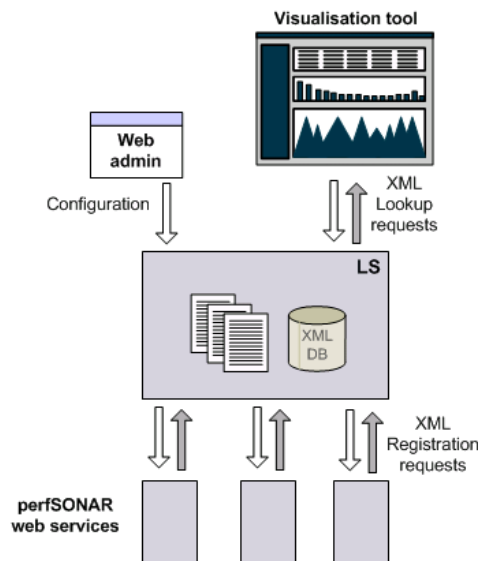


Fig. 2.3 Diagrama del funcionament del servei de *lookup* o d'anunci

Els servidors locals també s'actualitzen amb aquesta informació global per assegurar que un client que faci una consulta, rebi la informació de tots els serveis que hi ha actius fora del domini.

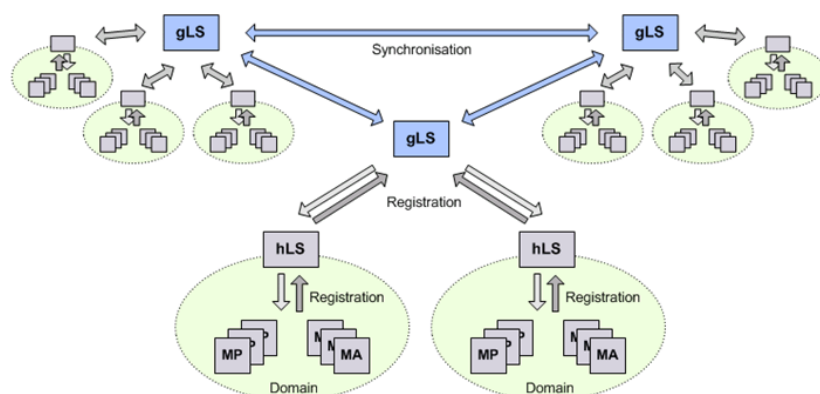


Fig. 2.4 Esquema del funcionament del registre i sincronització entre serveis de lookup locals i globals respectivament.

Un servei de perfSONAR degudament configurat, començarà a enviar missatges XML de registre al servei de *Lookup*. El servei de *Lookup* detecta doncs que un nou servei perfSONAR està disponible, i afegirà una entrada a la taula de la base de dades eXist indicant la disponibilitat d'aquest servei.

A la figura 2.5, mostrada més endavant, es pot observar l'entrada XML a la base de dades eXist en la que es mostren el registre d'un servei a un Lookup Service.

2.3.3. Els missatges de registre a serveis de *Lookup*

Per al registre d'un servei, es requereix certa informació obligatòria, però es pot incloure informació addicional que pugui ser interessant.

Els missatges de registre ha de constar de dues parts:

- Descripció del servei.
- Un camp de dades que contingui informació relacionada amb els continguts o capacitats del servei.

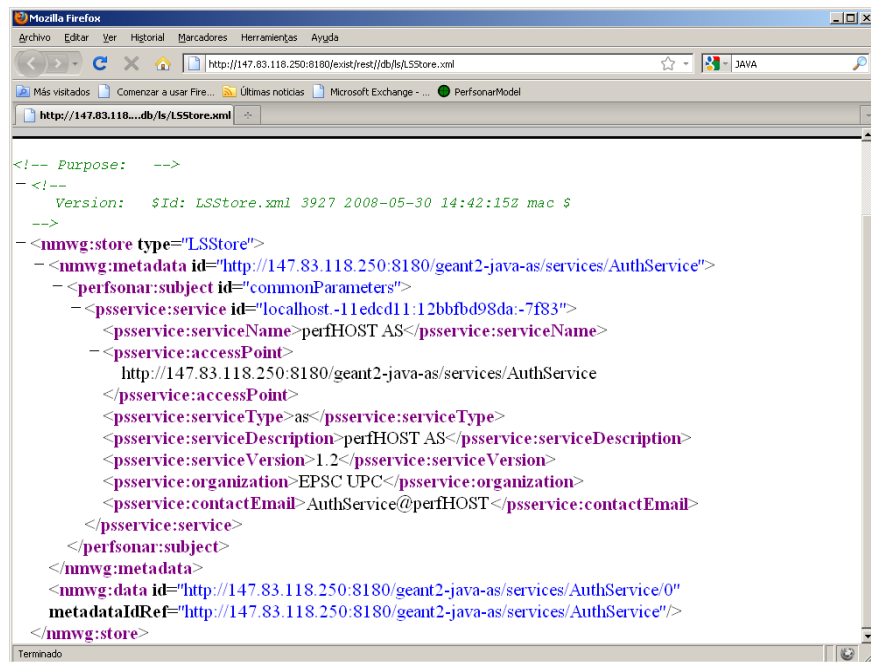


Fig. 2.5 Vista del registre d'un servei d'autenticació a la base de dades eXist del servei de Lookup

2.3.4. Problemes durant la instal·lació del servei

Durant la instal·lació dels prerequisits i del servei de *Lookup* no s'ha trobat cap problema que requereixi menció.

2.4 Authentication Service

El servei d'autenticació és l'encarregat de la validació en cas que aquesta sigui necessària per a l'accés a les dades que proporciona el sistema.

Donat que durant la realització d'aquest treball s'està estudiant el funcionament dels diferents serveis i les dades que contindran els serveis no comprometen cap sistema, s'opta per instal·lar aquest servei i configurar-lo però no s'aplicarà la validació per a cap missatge o servei.

La figura 2.7 mostra un diagrama del funcionament del servei d'autenticació.

Un usuari vol accedir, per exemple, a unes mesures de rendiment publicades a un servei de mesures. L'usuari es valida a partir de la perfSONAR UI contra un proveïdor d'identitats de Géant, a continuació la perfSONAR UI contacta amb el servei de mesures en qüestió i verifica els missatges que pot o no pot acceptar i així ho indica al servei, que serà l'encarregat d'entregar o no les dades.

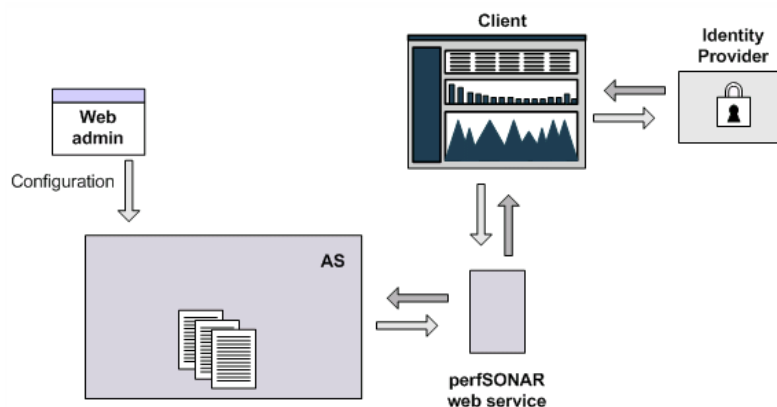


Fig. 2.7 Esquema del diàleg i elements que intervenen al realitzar una consulta que requereix autenticació

2.4.1 Sistema de validació

Per a la validació, des de Géant, s'ha creat un sistema de validació conjunt per a totes les entitats que hi participen, anomenat *Géant Identity Provider System*.

Aquest sistema apareix de forma temporal fins a la implantació total del sistema *eduGAIN*, que pretén compartir de forma correcta i segura les dades d'identitat entre les xarxes nacionals i la resta d'institucions. Actualment només les xarxes de recerca europees i altres institucions associades poden demanar un compte GIdP, i es necessita per a l'accés a dades protegides d'altres xarxes de recerca. En un futur serà possible aconseguir comptes d'altres proveïdors d'identitat.

Es presenten tres escenaris diferents per a la validació a l'entorn perfSONAR:

- **Client Automàtic:** Es dona quan el client no té cap interacció amb un usuari o humà, com podrien ser *scripts* PERL o PHP. Aquests *scripts* han de facilitar un certificat tipus X.509 juntament amb el missatge que fa la consulta.

- **Usuari darrere un client:** És el cas en que un usuari accedeix al sistema mitjançant una interfície gràfica com la perfSONAR UI sense capacitat de re-adreçament web. Un cop es connecta, el client ha d'enviar el missatge de sol·licitud incloent un certificat X.509 que s'obté a través de un servei SASL [15], ja que el certificat necessari depèn de l'usuari que hi hagi al darrere.
- **Client a un contenidor web:** Un usuari fa una petició a un servei, que alhora ha de fer petició a un altre. Un cop l'usuari fa el requeriment, el client web el redirecciona cap al sistema de validació de la universitat o centre validador de l'usuari, i el sistema de validació local respon al client web, que enviarà el missatge amb les credencials al sistema d'autenticació.

2.4.2 Prerequisites

Per a la instal·lació del servei d'autenticació, es requereix haver instal·lat amb anterioritat, els serveis que es mencionen a l'apartat 2.2:

La instal·lació i configuració del servei es presenten a l'annex F.

2.4.3 Recull de problemes durant la instal·lació del servei

Un cop realitzada la instal·lació com s'indica a l'annex F, al accedir a la *web-admin* per realitzar les proves de connexió, s'han trobat dos errors.

A un d'ells no s'ha trobat un motiu, però sí una solució. Després de la instal·lació es troba que al accedir a la *web-admin*, no s'ha trobat la plana habitual sinó una de diferent. Observant els arxius continguts als directoris del servei dins de l'estructura de directoris del Tomcat, es troba un arxiu '*index.html*' que no conté ordres per a l'execució de codi, per tant es reanomena com a '*index.html.old*' provocant que el servidor no utilitzi aquest arxiu com la porta d'entrada al servei. En aquest punt i després de reiniciar el servei Tomcat, el motor web utilitzarà l'arxiu '*index.jsp*' com a entrada del servei.

Aquí apareix el segon problema, ja que al carregar la pàgina, el servidor retorna un error de compilació de la plana *JSP*. El codi d'error complet es pot consultar a l'annex F,. Es decideix en aquest punt revisar el codi de l'arxiu i realitzar les modificacions que permetin la compilació de la plana *JSP*. Al mateix annex F es presenta un codi que permet l'execució de la plana sense problemes.

Consultant a la llista de correu es indiquen altres usuaris de perfSONAR, concretament de RedIris, que tenien el mateix problema [16] i es va indicar el mètode indicat per a la seva solució.

CAPITOL 3. ELS SERVEIS D'ARXIU DE MESURES

En aquest capítol es presenten els serveis que proporcionen arxius de mesura a l'entorn perfSONAR. Aquests arxius són els encarregats d'emmagatzemar diversos paràmetres de rendiment com utilització de la interfície, paquets descartats, nombre d'errors de recepció o nombre de paquets descartats a una interfície, d'aquelles interfícies que s'hi defineixin. El capítol comença explicant el servei d'arxius *Round Robin database* per a continuar comentant l'arxiu de mesura SQL.

3.1 RRD-MA

El '*Round Robin database Measurement Archive*' emmagatzema series temporals de dades, obtingudes a través d'eines de mesura SNMP, a bases de dades Round Robin, permetent l'accés a les dades des d'una consola perfSONAR UI.

Quan un usuari vol accedir a les dades arxivades, des de la perfSONAR UI s'envia un missatge XML al servei RRD MA, el servei processa la petició, realitza la consulta a la base de dades i s'encarrega de respondre via XML al client amb les dades sol·licitades.

A la figura 3.1 a continuació, es mostra un esquema de funcionament del servei.

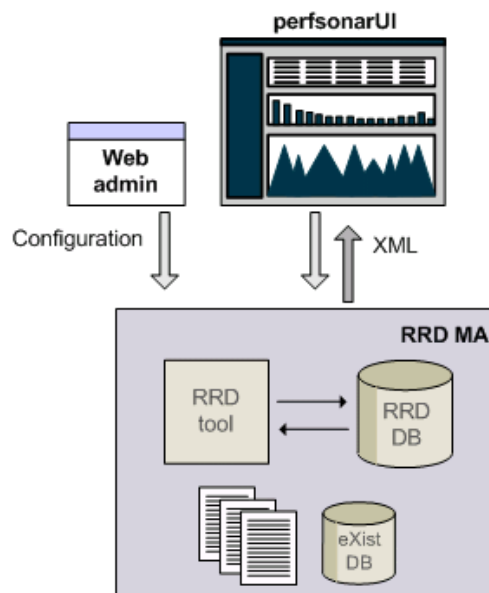


Fig. 3.1 Esquema del funcionament i serveis que intervenen al realitzar una consulta al servei RRD-MA de perfSONAR

3.1.1 Round Robin Database

Els arxius de bases de dades de tipus *Round Robin* [7], com qualsevol tipus de base de dades, desa i disposa una sèrie de valors que estan dintre d'un context, però cal destacar que aquest servei és orientat a l'emmagatzematge de dades basades en sèries temporals i alhora evitar que un creixement excessiu del nombre de bytes que ocuparan els elements al disc. Per això s'utilitza un esquema circular, on les diferents mesures emmagatzemades són sobreescrites un cop ha passat el període que es defineixi.

Suposem que estem monitoritzant els enllaços de sortida del CERN (*Tier-0* a *Tier-1*)¹. Aquest enllaç ha de realitzar la transferència de grans volums de dades. Per tant voldrem saber el valor de la utilització cada pocs segons per a detectar possibles fallides. Es defineix realitzar una mesura cada 5 segons, de manera que, en un dia tindriem:

$$\frac{1 \text{ mesura}}{5s} \times 60 \frac{s}{\text{min}} \times 60 \frac{\text{min}}{h} \times 24 \frac{h}{\text{dia}} = 17.280 \frac{\text{mesures}}{\text{dia}} \quad (3.1)$$

Sabent que existeixen onze enllaços *Tier-0 – Tier-1* obtenim doncs:

$$17.280 \frac{\text{mesures}}{\text{dia}} \times 11 \text{ enllaços} = 190.080 \frac{\text{mesures x enllaç}}{\text{dia}} \quad (3.2)$$

Ara bé, aquest tipus de sistema, requereix que es mostri correctament la informació actual, però no cal tanta precisió amb l'estat de la xarxa en un moment determinat mesos enrere. És necessari de conèixer aquesta informació però potser no ho és pas saber el valor exacte, sinó tan sols s'ha d'apreciar la anomalia si es que es va produir, i per tant no cal guardar la informació amb tant detall temporal.

Una base de dades RRD és capaç d'agrupar la informació en intervals de temps definits per l'usuari. Es pot definir que l'eina desi les dades amb una precisió de cinc segons per a la darrera hora, la unitat de temps entre proves definida anteriorment, un minut per a les següents vint-i-tres hores, deu minuts per a una setmana, i així successivament.

A continuació, a la figura 3.2, es mostra l'esquema d'una base de dades RRD que rep mesures cada cinc minuts i desa la informació en dos arxius, el primer d'una hora (12 mesures) i un altre amb un total de 24 mesures on cada resultat surt de fer la mitja de les mesures del primer arxiu després d'una hora. En aquest cas, quan s'hagin omplert els cercles del 9 al 12 del cercle interior, es farà la mitja dels valors i es desarà al cercle on apunta el punter diari. De manera que, cada dia que passi la base de dades haurà canviat per complet. El punter horari canviarà de posició cada cop que es rebí una mesura, cada cinc

¹ Es coneix com *Tier*, els diferents nivells de procés de dades que reben dades del projecte del LHC, de manera que el *Tier-0* seria a les instal·lacions del CERN a Ginebra, i el *Tier-1* seria un node que té connexió directa amb el node central. A Catalunya hi ha un *Tier-1* a les instal·lacions del Port d'Informació Científica (PIC), a la Universitat Autònoma de Bellaterra.

minuts, mentre que el punter diari ho farà cada hora a mida que es tinguin un total de dotze mesures noves.

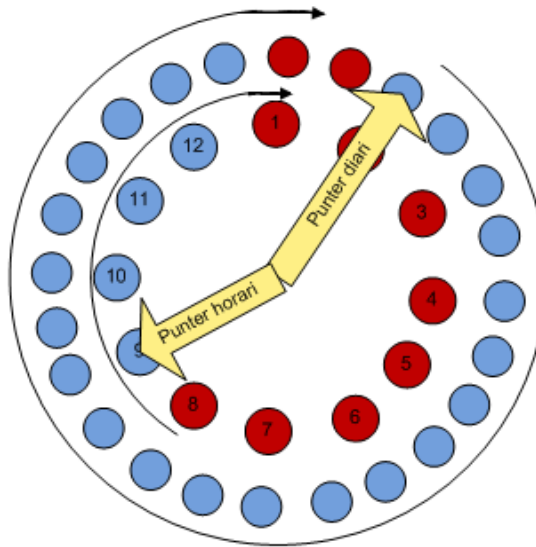


Fig. 3.2 Mostra del funcionament circular de les bases de dades tipus rrd, es mostra una base de dades amb un arxiu horari i diari.

3.1.2 Prerequisits del servei

Per a la instal·lació del servei s'han d'instal·lar components propis per a treballar amb bases de dades RRD.

Aquests prerequisits s'haurien d'instal·lar juntament amb el servei principal descrit en aquest apartat, ja que estan marcats com prerequisits al repositori de perfSONAR.

3.1.2.1 *RRDTool*

Aquesta és l'eina essencial per a realitzar consultes a aquest tipus de bases de dades, també crearà les bases de dades RRD i s'utilitzarà per a actualitzar els valors d'aquestes. Permet també entre d'altres funcions la creació de gràfiques, la generació de fitxers XML per a la realització de còpies de seguretat de les dades i la funció corresponent per a la recuperació d'aquestes.

3.1.2.2 *librrd2*

Aquest paquet conté les llibreries compartides per a la utilització de bases de dades RRD. Es recomana revisar l'annex G ja que durant la realització d'aquest TFC, aquest paquet ha estat retirat dels repositoris de Debian.

3.1.2.3 *RRDJTool*

En aquest cas el que s'instal·la és un conjunt de llibreries Java que ens permetran realitzar consultes a les bases de dades. Aquestes llibreries serveixen de passarel·la entre el servei Apache Tomcat i les eines RRD, per a fer consultes a les bases de dades.

A l'annex G es pot trobar el procés d'instal·lació i configuració del servei *RRD-MA*

3.1.3 Recull de problemes durant la instal·lació del servei

Durant la realització d'aquest TFC, un dels prerequisits del servei RRD-MA ha deixat d'estar disponible als repositoris de Debian, però continua marcat com a prerequisit als repositoris de perfSONAR, de manera que quan s'intenta realitzar la instal·lació, apareixen problemes.

Els següents paquets

- Geant2-java-rrd-ma
- Perfsonar-java-rrd-ma
- RRDjtool

tenen una dependència del paquet librrd2, de manera que si no s'instal·la aquest paquet, no es podran instal·lar la resta de paquets necessaris per al servei.

Es va consultar a la llista de correu de perfSONAR, i el responsable del desenvolupament del servei va recomanar la instal·lació de la propera versió, en fase de proves, on no s'hauria de donar aquesta dependència i per tant aquest problema s'hauria de solucionar.

A l'annex G s'indiquen les passes per a la instal·lació d'aquesta versió, que va resultar també fallida, ja que un dels paquets també depenia de la llibreria en qüestió.

Donat que es requeria per a la instal·lació del servei, i per a no alterar les instruccions d'instal·lació, es decideix localitzar i descarregar el paquet librrd2 des d'un servidor d'Internet, i publicar-lo a un repositori local creat per a l'ocasió. A l'annex G també es poden trobar les passes que es van seguir per a la publicació al repositori local. La publicació de la llibreria en el repositori local, va permetre una correcta instal·lació dels prerequisits i el propi servei.

3.1.4 L'arxiu de configuració

L'arxiu de configuració ha de contenir com a mínim:

1. Bloc de Meta-Dades:

- Meta-Data Id: Identificador del conjunt de dades que estem tractant.
- Hostname: El nom de l'equip sobre el que es realitzen les proves.
- ifAddress: El tipus de la interfície que es monitoritza i l'adreça.
- ifName: El nom d'aquesta interfície.
- EventType: Tipus de prova que s'hi ha realitzat.

2. Bloc de Dades:

- Data Id i Meta Data Id Ref: Identificador del bloc de dades i nom del bloc de meta-dades al qual s'associen les dades que estem definint.
- Paràmetres:
 - *File*: Ruta de l'arxiu RRD que conté les dades.
 - *DataSource*: Nom amb el qual es relacionen les dades que es volen proporcionar.
 - *valueUnits*: Unitats en les que es desen aquestes dades.
 - *EventType*: L'esquema al qual pertany la prova definida.

A l'hora de generar l'arxiu de configuració cal tindre en compte una limitació clau per al funcionament del servei: cada bloc de Meta-dades només pot apuntar a una font de dades (o columna) d'una base de dades RRD.

Això implica que, en el cas de que estiguem monitoritzant la utilització d'entrada i sortida d'una mateixa interfície i desant les dades a una base de dades amb dues fonts de dades, s'hauran de definir dos blocs de meta-dades, un per a la utilització d'entrada i un altre per a la utilització de sortida.

Ara bé, també cal indicar que no hi ha una limitació del nombre de blocs de meta-dades i dades que es puguin definir. Així mateix, destaquem que cada canvi de la topologia de la xarxa monitoritzada ha de ser modificat a l'arxiu de configuració, ja que com es pot observar, el sistema requereix d'aquest arxiu per a mostrar la informació que es vol mostrar.

La creació d'aquest tipus d'arxius en xarxes molt complexes pot suposar una bona feina, així que, perfSONAR proporciona dintre de la pròpia distribució un script que genera l'estructura comentada anteriorment. Mitjançant la utilització d'aquest codi i algun altre codi que desenvolupi l'administrador del servei, es podria omplir de forma semiautomàtica els blocs corresponents.

L'annex G conté més informació sobre l'arxiu de configuració.

3.1.5 Obtenció de dades

Per a la obtenció de les dades, com és habitual en la majoria de sistemes de monitorització, s'utilitza el servei *Simple Network Management Protocol* (SNMP), amb el qual es sol·licita a l'equip els paràmetres que ens interessa monitoritzar.

S'ha instal·lat i configurat el servei SNMP a les màquines utilitzades de manera que es poden obtenir els valors desitjats a partir de les sentències que es mostren a continuació:

- Octets d'entrada:

```
# snmpwalk -Ov -v2c -c perfhost 147.83.118.251  
interfaces.ifTable.ifEntry.ifInOctets.2
```

- Octets de sortida:

```
# snmpwalk -Ov -v2c -c perfhost 147.83.118.251  
interfaces.ifTable.ifEntry.ifOutOctets.2
```

- Errors de recepció a la interfície:

```
# snmpwalk -Ov -v2c -c perfhost 147.83.118.251  
interfaces.ifTable.ifEntry.ifInErrors.2
```

- Errors de transmissió a la interfície:

```
# snmpwalk -Ov -v2c -c perfhost 147.83.118.251  
interfaces.ifTable.ifEntry.ifInDiscards.2
```

A continuació, el script col·lector de dades processa el resultat de la consulta i desa el valor concret en una variable i disponible per a la seva publicació a la base de dades.

3.1.6 Mètode alternatiu de publicació de dades

Degut a que a la guia d'administració [9] no es comenta cap mètode per a la obtenció ni publicació de dades, hem desenvolupat uns *scripts* que permeten la captació i publicació de les dades al sistema perfSONAR. Anomenem aquest mètode com "alternatiu" ja que durant la realització d'aquesta memòria es va trobar un altre mètode que permet la creació i publicació de dades de forma automàtica. El mètode 'oficial' es pot consultar al següent punt d'aquest mateix document.

Es descriu a continuació les sentències que ha d'executar el *script* captador de dades per a la publicació d'aquestes a les bases de dades RRD.

La publicació a una base de dades RRD es realitza també amb l'eina *rrdtool*, però utilitzant en aquest cas l'opció 'update'. Per a emmagatzemar les dades a l'arxiu, s'han d'indicar els següents paràmetres:

- Ruta i/o nom de la base de dades.
- Valor horari del moment al que corresponen les dades. Si les dades són d'aquest moment es pot utilitzar el comodí 'N'.
- Valors de les fonts de dades.

Per exemple, suposant que estem al directori on hi ha la base de dades, s'hauria d'executar una sentència com la mostrada a continuació:

```
# rrdtool update ./db.rrd N:[valor_font1]:[valor_font2]
```

A l'annex G es pot trobar el codi sencer del script col·lector de dades.

3.1.7 Publicació de dades

Durant la realització d'aquest document es van trobar uns arxius que contenen un tipus de missatge que pot rebre el servei per a una publicació 'semiautomàtica' de dades.

Es va demanar a la llista de correu de perfSONAR si hi havia algun mètode per a realitzar aquestes accions, donat que resultava molt estrany que un sistema de monitorització no implementi aquesta funció, no obtenint resposta de cap dels membres de la llista [17]. A la guia d'administració del servei tampoc es troba cap mètode per a fer-ho així que es va optar per generar *scripts shell* que fessin aquestes tasques a partir de la utilització dels components que s'instal·len amb el servei.

Per a que el servei accepti aquest tipus de missatges cal accedir a la plana de configuració avançada del servei a partir del següent enllaç:

<http://147.83.118.251:8180/geant2-java-rrd-ma/ServiceAdmin?serviceProperties>

I definir la opció:

```
service.ma.xmlldb.db_store → 'on'
```

Un cop reiniciat el servei s'intenta enviar el missatge XML corresponent i el servei RRD-MA contesta indicant que no s'ha afegit la informació a la base de dades per no disposar de permisos d'escriptura al directori:

```
/var/run
```

S'aprecia però que a la configuració no es defineix aquest directori per a desar bases de dades, sinó el directori:

```
/usr/lib/perfsonar/services/geant2-java-rrd-ma/WEB-INF/classes/perfsonar/conf/
```

Un cop es donen permisos d'escriptura al servei tomcat al directori que ens indica al primer missatge (veure punt 3.1.8, modificació de permisos), el servei respon amb un missatge indicant que s'ha desat el valor a una base de dades que s'ha generat, confirmant així que perfSONAR contempla aquesta opció. Suposem que no s'explica com a tal ja que la publicació de dades es pot realitzar d'altres maneres que no afecten al funcionament del servei i que semblen més senzilles.

Donada aquesta troballa, es decideix generar un codi *Java* que permeti la publicació de les dades a través del servei perfSONAR. Es crea una classe inicial que connecta amb el servei utilitzant les classes de connexió HTTP del paquet *java.net*. Però el missatge de resposta del servei indica que el missatge que s'ha enviat és erroni.

Es realitza en aquest punt una captura de tràfic mitjançant el software *wireshark* [19], i es determina que la perfSONAR UI encapsula el tràfic en un 'sobre' SOAP mentre que el codi generat estava enviant el missatge XML sense aquestes capçaleres.

Així doncs es decideix modificar el programa per a que generi una connexió TCP a través d'un *socket* a la URL d'entrada del servei i enviar tots els missatges HTTP i XML directament a través d'aquest *socket*. D'aquesta manera, el codi envia el missatge XML correctament encapsulat en un 'sobre' SOAP.

Tot i que el missatge està correctament encapsulat, el servei continua responnent amb missatges d'error, aquests errors es deuen a la manera en la que va encapsulat el missatge XML, ja que cal tabular correctament el missatge XML així com els salts de línia entre les diferents etiquetes.

Mitjançant unes proves realitzades es determina que aquest sistema desa durant un dia a la mateixa base de dades, però un cop passa aquest dia, el sistema no genera un nou arxiu, ni continua desant valors a la mateixa base de dades, no podent determinar el motiu d'aquest comportament.

El codi utilitzat per a la publicació de dades a través del servei RRD-MA es pot consultar a l'annex H.

3.1.8 Revisió de permisos

Per a un correcte funcionament del sistema, cal una verificació dels permisos dels directoris on es desen els arxius de bases de dades així com els arxius de configuració. D'aquesta manera ens assegurem que el servei Apache Tomcat podrà realitzar les accions que correspongui. Es recomana la verificació de la següent ruta, que s'utilitza per defecte per als arxius de configuració i en la que també hi podem desar la base de dades RRD:

```
/usr/lib/perfsonar/services/geant2-java-rrd-ma/WEB-INF/classes/perfsonar/conf/
```

Caldrà utilitzar la sentència:

```
# chown tomcat55:root ./*
```

Per a permetre a l'usuari tomcat55, l'execució de codi des del servei Apache Tomcat i assignant el grup 'root' a tots els arxius que hi hagi al directori.

3.2 El SQL-MA

El servei d'arxius de mesura en SQL utilitza el motor MySQL per a l'emmagatzematge i la consulta de dades.

La figura 3.3, mostra l'esquema de funcionament d'aquest servei:

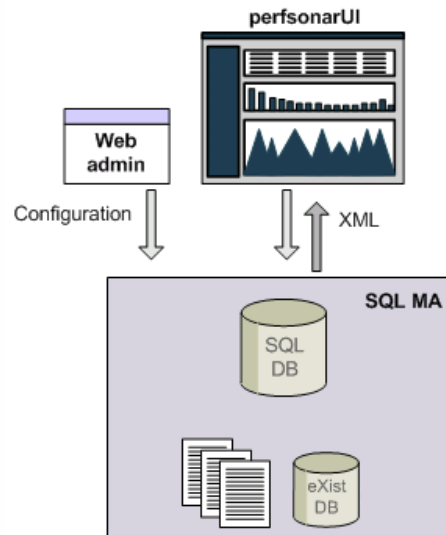


Fig. 3.3 Esquema de la connexió i dels components que intervenen en una consulta SQL MA

De la mateixa manera que en el servei RRD-MA, un client des de la interfície gràfica de perfSONAR es connecta a la base de dades eXist del servei SQL-MA en la que es defineixen les interfícies monitoritzades, un cop es seleccionen les dades que es volen obtenir, el servidor estableix connexió amb el motor de base de dades i les fa arribar a la consola per a que es mostrin.

El sistema permet l'emmagatzematge i la publicació de les següents mesures:

- Utilització
- Capacitat d'interfície
- Nombre de paquets descartats per una interfície IP
- Nombre d'errors de recepció d'una interfície IP
- Paquets descartats
- Estat de circuit
- *Throughput* assolible
- *Throughput* UDP

3.2.1 Prerequisits

Previ a la instal·lació del servei, s'ha d'instal·lar el servei MySQL versió 5, assumint que ja s'han instal·lat els prerequisits del sistema. La instal·lació del

component s'executa mitjançant la següent comanda amb privilegis d'administrador.

- `apt-get install mysql-server mysql-client-5.1`

Durant la instal·lació del motor MySQL, es sol·licitarà un password d'administrador del servei de base de dades.

La resta de comentaris sobre la instal·lació i la configuració del servei es pot trobar a l'annex I.

3.2.2 Preparació del servei MySQL

Aquest servei està programat per a què en cas que l'implementador ja disposi de dades del mateix tipus emmagatzemades, pugui adaptar els arxius de configuració per a extraure les dades correctament des del seu sistema, tal i com es comenta a l'apartat corresponent a [9]

En el nostre cas, al no tenir una estructura prèvia, s'opta per seguir les recomanacions del servei, creant una estructura de dades predefinida.

Per a crear aquesta estructura, s'ha d'executar un *script* `.sql` amb el client de connexió a la base de dades que conté la estructura a crear en el motor. Aquest *script* conté les sentències necessàries per a la creació de les bases de dades i taules i que s'executa amb la següent sentència de comandes, que sol·licitarà el *password* d'administrador del motor SQL definit a l'apartat 3.2.1.

- `mysql -u root -p < /usr/lib/perfsonar/services/geant2-java-sql-ma/WEB-INF/classes/perfsonar/conf/mysql-sqlma-dbsetup.sql`

Un cop finalitzat, es crea una estructura amb les taules que apareixen la figura 3.4, més endavant.

3.2.3 Obtenció de dades

S'ha utilitzat en aquest punt el mateix sistema de captació de dades que es defineix a l'apartat corresponent del servei RRD-MA.

perfsonar_ma.interdomain_link id : varchar(255) time : timestamp status_oper : varchar(255) status_admin : varchar(255) alarm_id_set : varchar(255) comments : varchar(255)	perfsonar_ma.perfsonar_discards metadataId : varchar(30) value : float(16,4) valueUnits : varchar(30) timeValue : int(11) ts : timestamp	perfsonar_ma.perfsonar_links id : varchar(255) time : timestamp status_oper : varchar(255) status_admin : varchar(255) alarm_id_set : varchar(255) comments : varchar(255)
perfsonar_ma.perfsonar_errors metadataId : varchar(30) value : float(16,4) valueUnits : varchar(30) timeValue : int(11) ts : timestamp	perfsonar_ma.domain_link id : varchar(255) time : timestamp status_oper : varchar(255) status_admin : varchar(255) alarm_id_set : varchar(255) comments : varchar(255)	
perfsonar_ma.perfsonar_utilization metadataId : varchar(30) value : float(16,4) valueUnits : varchar(30) timeValue : int(11) ts : timestamp	perfsonar_ma.perfsonar_iperf metadataId : varchar(30) value : float(16,4) valueUnits : varchar(30) throughput : float(16,4) jitter : float(16,4) loss : varchar(15) numBytes : decimal(40,0) numBytesUnits : varchar(15) interval : varchar(30) time : int(11) format : varchar(30) protocol : varchar(10) bufferSize : int(11) windowSize : int(11) windowSizeRequest : varchar(30) timeDuration : varchar(30) arguments : varchar(50) test_session_id : varchar(30) ts : timestamp	
perfsonar_ma.cmp_ping metadataId : varchar(255) seqNum : int(10) unsigned value : float(16,4) valueUnits : varchar(4) ttl : tinyint(3) unsigned numBytes : int(11) numBytesUnits : varchar(10) timeValue : mediumtext timeType : varchar(4) ts : timestamp		

Fig. 3.4 Estructura de les taules SQL proposada per perfSONAR per al servei SQL-MA

3.2.4 Publicació de dades

Per a la publicació de dades, s'ha utilitzat un programa propi en codi Java que realitza el càlcul de la utilització en bits/segons (bps). La utilització es pot entendre com:

- Percentatge del temps que hi ha activitat en una interfície.
- Nombre de dades transmesa per unitat de temps.

En el sistema perfSONAR la utilització s'entén de la segona forma, de manera similar a l'ample de banda, ja que tant a la definició de les unitats com a la representació d'aquestes a la interfície gràfica, s'indica que les unitats son Mbps o bps.

L'*script* de captació de dades ens facilita la lectura SNMP actual del nombre d'octets d'entrada o sortida, però per a calcular la utilització es requereix un valor de referència, el valor de la mesura anterior. Per tant es crea una nova taula a la base de dades, que contingui el valor anterior a la mesura que rebí el programa.

El codi del programa resultant es pot consultar a l'annex J

Un cop s'executa el programa de càrrega, rep per paràmetre el valor de les lectures SNMP actuals, els desa en variables i consulta a la taula creada quins son els valors de la mesura anterior.

Els comptadors SNMP són comptadors de trenta-dos bits que emmagatzemen el nombre d'octets que es reben o s'envien per la interfície, per tant quan el valor no es pot expressar amb una xifra de trenta-dos bits, comença a comptar des de zero. El programa verifica si el comptador ha donat la volta i posteriorment realitzarà el càlcul del nombre d'octets que s'han transmès per la interfície en els darrers cinc minuts per a realitzar el càlcul de la utilització mitjançant la següent formula:

$$\frac{\text{OctetsEntrada} + \text{OctetsSortida}}{\text{IntervalMesures}} \frac{(\text{Octets})}{(s)} \times \frac{8 \text{ (bits)}}{1 \text{ (Octet)}} = \text{Utilització} \frac{\text{bits}}{s} \quad (3.3)$$

on el valor de l'interval entre mesures, que s'ha comentat prèviament, és de 300 segons (5 minuts).

3.2.5 Arxiu de configuració del servei

El servei SQL-MA també requereix d'un arxiu de configuració que es desa a la base de dades eXist i que conté la definició dels blocs de dades i meta-dades que conformen les interfícies que s'estan monitoritzant. Aquesta definició és molt semblant a la descrita a l'apartat corresponent al servei RRD-MA, amb diferències als blocs de dades.

A continuació es mostra un fragment de codi d'exemple, del fragment que presenta diferències respecte al servei anterior:

```
<nmwg:data id="data1-test" metadataIdRef="meta1-test">
  <nmwg:key>
    <nmwg:parameters>
      <nmwg:parameter name="metadataId">meta1-test</nmwg:parameter>
      <nmwg:parameter name="ibatisConfig">ibatis-SqlMapConfig-utilization.xml</nmwg:parameter>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/characteristic/utilization/2.0
      </nmwg:parameter>
    </nmwg:parameters>
  </nmwg:key>
</nmwg:data>
```

En el servei anterior s'especificava en una de les línies d'aquest arxiu la ruta de la base de dades RRD, mentre que en aquest cas s'indica el fitxer de configuració *ibatis*, mitjançant el qual s'efectua la connexió al motor de bases de dades i conté informació sobre l'estructura de les bases de dades al sistema.

El següent apartat descriu en termes generals, el funcionament de l'eina Ibatis.

3.2.6 Passarel·la Ibatis

Ibatis, des de Juny de 2010 [20], anomenada MyBatis [21], és una eina que s'encarrega de relacionar objectes. En el cas de perfSONAR relaciona arxius XML amb instruccions SQL.

D'aquesta manera, quan el servei SQL-MA rep una petició de dades, consulta a l'arxiu de configuració quin és l'arxiu Ibatis associat i aquest arxiu serveix de passarel·la per a traduir els valors de la interfície que es vol consultar, en consultes per a les taules SQL.

Un cop s'obtenen les dades, el servei les tradueix en els missatges XML que correspongui per a acabar-ho enviant a la interfície gràfica, encarregada de mostrar les dades.

Amb la modificació d'aquests arxius, un usuari que ja disposi de dades similars emmagatzemades a una estructura SQL diferent de la que proposa perfSONAR, podria adaptar el servei SQL-MA per a què realitzi les consultes a la seva base de dades.

3.2.7 Dificultats trobades durant la instal·lació del servei

No s'han trobat dificultats remarcables durant la instal·lació del servei, trobant-ne però durant la configuració i les proves de funcionament.

Un cop realitzades les accions explicades en els punts anteriors, el servei no respon amb les dades de rendiment, apareixen problemes de connexió amb la base de dades i de configuració del propi servei.

Per tant s'opta per demanar ajuda a la llista de correu de perfSONAR, i en comunicació amb el Sr. Piotr Pikusa [22], del centre de supercomputació i de xarxes de Poznan, ens indica diferents proves que es poden realitzar per a verificar el funcionament del servei.

Després de diversos correus, el servei es capaç de comunicar amb la base de dades eXist i enviar les dades de les interfícies a la consola, però no s'han pogut rebre les dades de la interfície.

CAPITOL 4. ELS PUNTS DE MESURA

En aquest capítol es mostrarà el funcionament dels dos sistemes de mesura que proporciona perfSONAR. El primer, el *BWCTL*, permet l'execució de proves d'ample de banda sota demanda, amb els servidors disponibles. Després s'indica el procés d'instal·lació del servei *Telnet/SSH*, que té com a objectiu, entre d'altres opcions, la mostra de paràmetres de configuració de dispositius de xarxes remotes.

4.1. EI *BWCTL-MP*

El *BWCTL-MP* és un servei de mesura capaç d'efectuar proves d'ample de banda sota demanda, entre dos equips. A Internet hi ha diversos serveis que permeten fer proves d'ample de banda entre el servei i l'equip de l'usuari, però els resultats que s'obtenen a partir d'aquestes proves són vàlids per al camí entre l'equip i el servei d'internet de manera que només es pot determinar si hi ha algun problema amb la connexió local. Amb la implementació del servei *BWCTL-MP* a dues organitzacions, es poden realitzar proves d'ample de banda disponible a través del camí que seguiria una transmissió entre elles.

La figura 4.1 mostra un esquema del funcionament del servei *BWCTL-MP*.

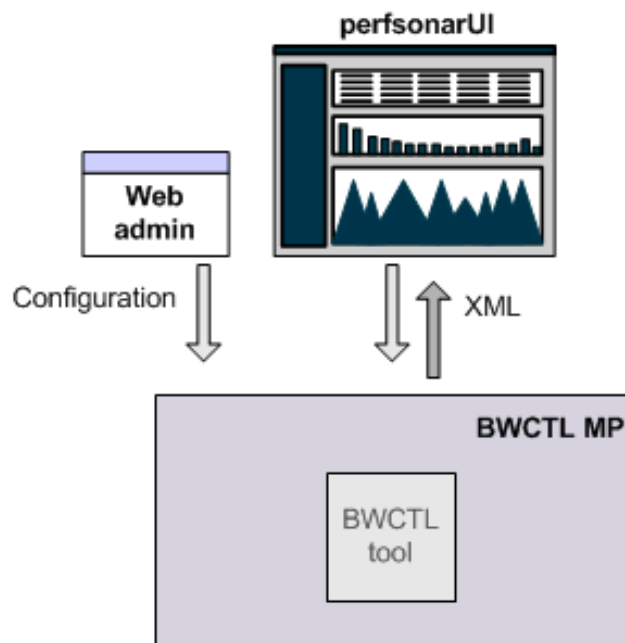


Fig 4.1 Funcionament del servei *BWCTL* al realitzar una prova d'ample de banda des de la *perfSONAR UI*

4.1.1 Prerequisits

4.1.1.1. NTP

El servei BWCTL requereix sincronització horària del servidor, de manera que s'ha d'instal·lar el servei NTP (*Network Time Protocol*) [23], que s'encarrega de sincronitzar l'hora del servidor local amb servidors NTP disponibles a Internet.

4.1.1.2. Iperf

Iperf [24] és un software que permet l'execució de proves de mesura del màxim ample de banda entre dos extrems, permetent a l'usuari indicar determinats valors i característiques de la comunicació. Un cop s'ha efectuat la prova, l'eina informa de diversos paràmetres com l'ample de banda assolit, el retard i la seva variança (*jitter*), o la pèrdua de paquets.

4.1.1.3. Apache Web Server

Segons s'indica a [9], cal instal·lar aquest servei per a poder utilitzar la interfície *web-admin*, tot i que degut a certes variacions del servei trobem que no és del tot necessari la instal·lació d'aquest servei. De totes maneres, comentem la instal·lació i configuració del servei al annex K.

4.1.1.4. Bandwith Test controller

El BandWith Test Controller (BWCTL) [26] funciona posant en contacte dos processos *bwctld* a cadascuna de les xarxes entre les que volem efectuar la prova. En el moment de l'execució, el propi procés s'encarrega de controlar i assignar els recursos dels extrems.

Sense un servei d'aquest tipus, un usuari ha de requerir dels administradors de xarxa per a realitzar determinades proves de rendiment. El que es pretén amb aquest servei és tot el contrari: el servei per si mateix pot determinar i reservar els recursos necessaris permetent al propi usuari la realització de la prova. Un cop BWCTL rep una sol·licitud de mesura, es posa en contacte amb l'altre extrem de la connexió per a verificar la seva disponibilitat. El servei remot indica si requereix d'un temps d'espera abans de realitzar la prova i en cas de ser així es sincronitzen per realitzar-la tan bon punt sigui possible.

4.1.2 Problemes durant la instal·lació del servei

A banda del problema comentat amb la disponibilitat dels paquets als repositoris de perfSONAR, s'han trobat dos altres problemes durant la instal·lació del servei que s'ha realitzat tal i com s'indica l'annex K.

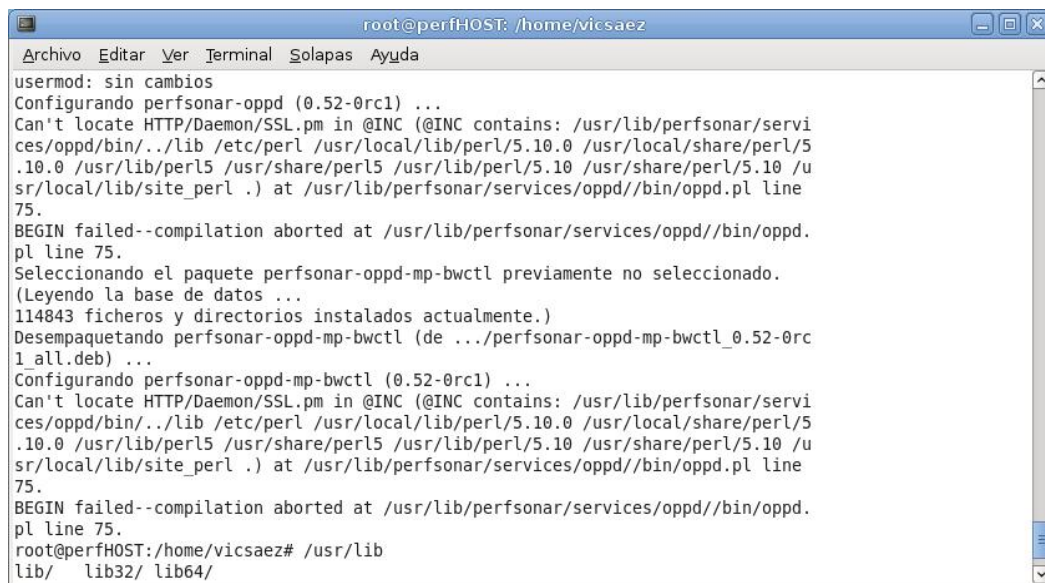
El primer apareix en el moment en el que finalitza la instal·lació i s'intenta executar l'arxiu `oppd.pl`, el servei indica que requereix d'unes llibreries Perl que no han estat instal·lades amb la resta dels serveis, de manera que cal executar la comanda que s'indica a continuació amb permisos d'administrador:

```
# apt-get install libipc-run-perl
```

Un cop s'acaba la instal·lació, s'intenta arrencar el servei amb la següent comanda:

```
# perl /usr/lib/perfsonar/services/oppd/bin/oppd.pl
```

Però, el sistema ens indica un altre error que es mostra a la figura 4.2 que es mostra tot seguit.



```
root@perfHOST: /home/vicsaez
Archivo Editar Ver Terminal Solapas Ayuda
usermod: sin cambios
Configurando perfsonar-oppd (0.52-0rc1) ...
Can't locate HTTP/Daemon/SSL.pm in @INC (@INC contains: /usr/lib/perfsonar/services/oppd/bin/./lib /etc/perl /usr/local/lib/perl/5.10.0 /usr/local/share/perl/5.10.0 /usr/lib/perl5 /usr/share/perl5 /usr/lib/perl/5.10 /usr/share/perl/5.10 /usr/local/lib/site_perl.) at /usr/lib/perfsonar/services/oppd/bin/oppd.pl line 75.
BEGIN failed--compilation aborted at /usr/lib/perfsonar/services/oppd/bin/oppd.pl line 75.
Seleccionando el paquete perfsonar-oppd-mp-bwctl previamente no seleccionado.
(Leyendo la base de datos ...)
114843 ficheros y directorios instalados actualmente.)
Desempaquetando perfsonar-oppd-mp-bwctl (de .../perfsonar-oppd-mp-bwctl_0.52-0rc1_all.deb) ...
Configurando perfsonar-oppd-mp-bwctl (0.52-0rc1) ...
Can't locate HTTP/Daemon/SSL.pm in @INC (@INC contains: /usr/lib/perfsonar/services/oppd/bin/./lib /etc/perl /usr/local/lib/perl/5.10.0 /usr/local/share/perl/5.10.0 /usr/lib/perl5 /usr/share/perl5 /usr/lib/perl/5.10 /usr/share/perl/5.10 /usr/local/lib/site_perl.) at /usr/lib/perfsonar/services/oppd/bin/oppd.pl line 75.
BEGIN failed--compilation aborted at /usr/lib/perfsonar/services/oppd/bin/oppd.pl line 75.
root@perfHOST:/home/vicsaez# /usr/lib
lib/ lib32/ lib64/
```

Fig. 4.2 Detall de la consola del perfHOST on es mostra un error al llançar el servei OPD.

A l'arxiu `oppd.perl` s'hi defineix una instància a la classe `HTTP::Daemon::SSL`, que crea un socket HTTP/1.1 per a peticions segures entrants.

S'entén que les connexions segures no han de ser un requisit per al correcte funcionament del servei, de manera que s'opta per comentar la línia 75 que s'indica com a errònia al terminal i s'intenta executar el servei amb la comanda anteriorment indicada.

Després d'arrancar el servei amb la comanda mencionada en aquest mateix apartat, torna aparèixer el *prompt*, indicant que el servei s'ha arrancat i es pot confirmar amb la comanda *ps*.

4.1.3 Verificació del funcionament del servei

Per verificar el correcte funcionament del servei de mesures BWCTL MP utilitzem la pestanya BWCTL la interfície gràfica perfSONAR UI, on cal seleccionar un *BWCTL Source* i un *BWCTL Destination* al menú *Query Options* a la part esquerra de la pantalla.

Un cop seleccionats el nostre servei com a *Source* i un qualsevol de la llista com a *Destination*, es poden seleccionar diferents paràmetres de la prova que es realitzarà com la durada i l'interval d'informació. A continuació cal prémer la opció *Query* del menú *Execute Query* per a començar la prova.

La imatge 4.3 a continuació mostra el resultat d'un test amb una durada (la màxima permesa) de 60 segons i un interval d'informació de 10 segons.

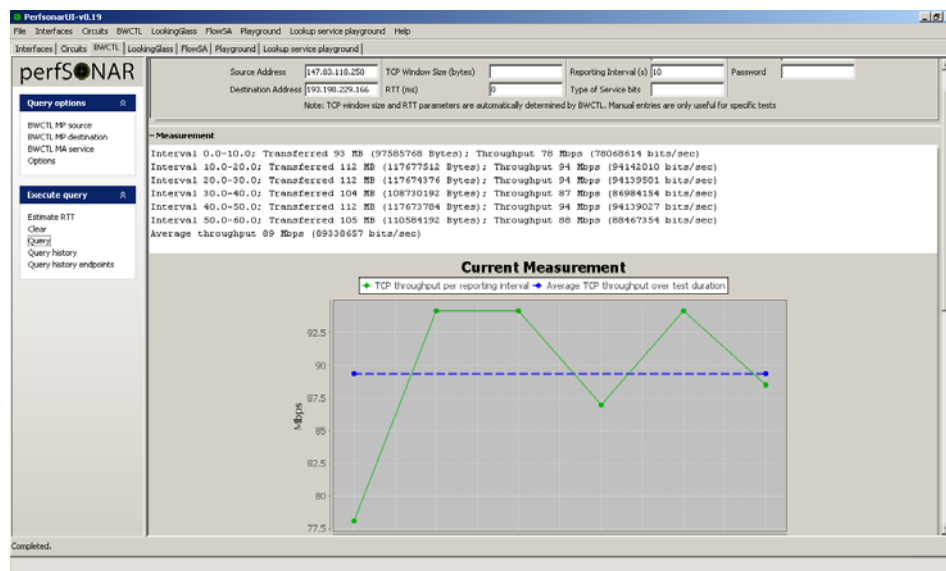


Fig. 4.3 Aspecte de la interfície perfSONAR UI després de la realització d'una prova BWCTL

Un cop finalitzada la prova, els resultats es mostren a la interfície gràfica a l'apartat *Results* com a text i a la gràfica de la part inferior.

4.1.4 Proves realitzades amb el servei BWCTL-MP

Amb l'ànim de provar el funcionament d'aquesta eina, es creen un *scripts* que executen proves d'ample de banda amb diversos servidors BWCTL disponibles i que s'obtenen a partir de la perfSONAR UI, a través d'una comunicació personal amb un membre d'Internet2 [30] i a partir de [31].

La finalitat d'aquestes proves es analitzar l'ample de banda resultant al realitzar la mateixa prova *bwctl* entre el nostre servei i d'altres serveis repartits geogràficament. Aquests resultats es poden consultar a l'annex K.

Donat que algunes proves s'han realitzat en horaris de baixa càrrega de xarxa, s'ha observat que en alguns casos, els valors més alts son propers a 100 Mbps, i és molt possible que la xarxa del laboratori sigui una limitació pels valors obtinguts. Suposem que el servei està pensat per a estar connectat a xarxes que disposin d'enllaços Gigabit Ethernet o enllaços més amb més capacitat del que es disposa al laboratori i que permetrà que els resultats del test siguin més fidels a la capacitat dels enllaços entre NREN's.

A [32] s'indiquen diverses opcions per a millorar els resultats obtinguts al realitzar aquest tipus de proves mitjançant la modificació de la mida de la finestra de transmissió i recepció del protocol TCP del sistema operatiu. Aquestes modificacions poden tenir importància si es disposa d'una connexió amb gran ample de banda, al laboratori C4-325, com hem comentat, es disposa d'una xarxa a 100Mbps en la que no s'han apreciat grans diferències al variar el valor d'aquests paràmetres.

4.2. El Telnet/SSH

El servei *Telnet/SSH*, permet als usuaris sol·licitar el *RRT*, *Round-Trip delay Time*, entre dues xarxes, mostrar comandes de configuració o realitzar comandes *Traceroute* per a observar el camí que segueix la transmissió entre dues xarxes. Un cop rebuda una sol·licitud, que es fa a partir de la interfície perfSONAR UI, el servei obté la informació dels *routers* a consultar i retorna la informació al client mitjançant un missatge XML.

A la figura 4.4 es mostra un diagrama de funcionament del servei.

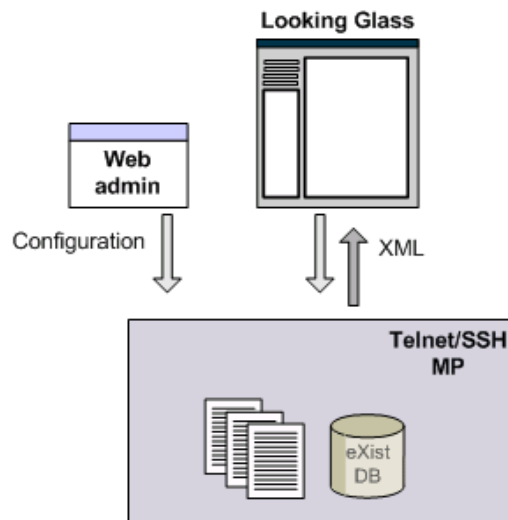


Fig. 4.4 Diagrama del funcionament del servei Telnet/SSH

4.2.1. Prerequisits

A banda dels prerequisits habituals de qualsevol servei de perfSONAR es requereixen també dispositius de xarxa que implementin el protocol SSH/Telnet, ja que per a obtenir les comandes s'utilitzen aquest tipus de missatges per a la transmissió d'ordres i per a la recepció de les sortides. La majoria de dispositius Cisco permeten la utilització d'aquest tipus de missatges.

4.2.2. Comprovació del servei

Per a comprovar que el resultat de la instal·lació s'ha d'accedir a la següent URL:

<http://147.83.118.250:8180/ps-mdm-sshtelnet-mp>

I seleccionant també l'opció *Service Test*, tal com s'ha detallat, per exemple, a l'apartat D.2.

4.2.3. Recull de problemes durant la instal·lació del servei

Durant la instal·lació tal i com s'indica al annex L d'aquest servei ens trobem amb un error de compilació de la plana d'accés al servei. La traça d'error que retorna el servei Tomcat, és la mateixa que es descriu al punt F.3. Realitzant les mateixes modificacions al codi que es van realitzar també en el cas del servei AS s'aconsegueix mostrar la plana correctament.

CAPITOL 5. LA INTERFÍCIE GRÀFICA DE PERFSONAR

perfSONAR proporciona una interfície gràfica, l'eina perfSONAR UI (*User Interface*), des de la qual un usuari pot realitzar proves i sol·licitar dades a d'altres serveis perfSONAR. Aquest capítol comença explicant el funcionament de les de la interfície gràfica de perfSONAR i les eines que proporciona. A la segona part es presenta una guia per a la implementació de noves funcionalitats per aquesta eina.

5.1 Instal·lació de l'eina

L'eina perfSONAR UI està disponible per a tots els sistemes operatius, ja que està programada utilitzant les característiques de Java, que s'han comentat prèviament. L'eina requereix d'una màquina virtual de Java instal·lada al sistema. En funció de la versió de l'eina es requerirà:

- Fins a la versió 0.08: Java Runtime Enviroment Standard Edition 1.4.
- A partir de la versió 0.09: Java Runtime Enviroment Standard Edition 1.5

Actualment la instal·lació de l'eina es realitza utilitzant els avantatges de *Java Web Start* [33], una funcionalitat que s'instal·la juntament amb la versió 1.5 de la màquina virtual de Java i que permet la descàrrega i execució d'aplicacions Java des de la web. És a dir, que es descarrega un petit programa que s'encarregarà de descarregar els arxius necessaris per la instal·lació i els instal·larà. Aquesta tecnologia facilita també la distribució d'actualitzacions, ja que en el moment d'arrencar l'aplicació, aquesta verifica si hi ha noves versions.

La darrera versió de la interfície perfSONAR UI està disponible per a descàrrega des de l'enllaç que es mostra a continuació:

<http://iris.acad.bg/perfsonar/perfsonar.jnlp>

Un cop s'executa l'arxiu *.jnlp*, l'aplicació comença el procés de descàrrega i d'instal·lació. Un cop finalitzat, s'executa l'aplicació com es mostra a la figura 5.1 a continuació.

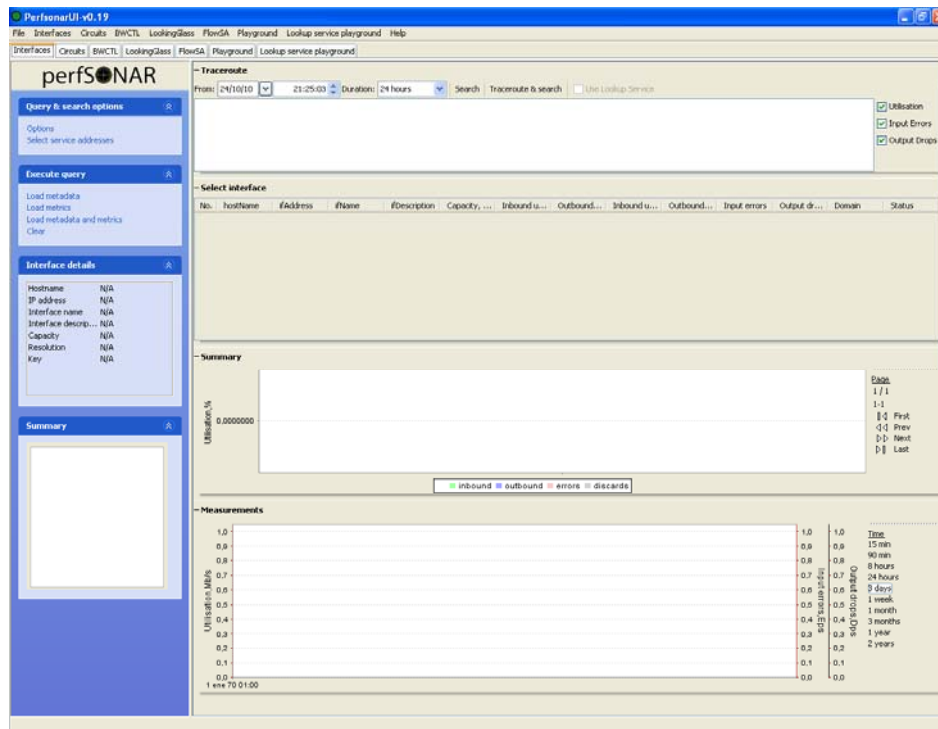


Fig. 5.1 Aspecte de la interfície gràfica perfSONAR UI

5.2 Funcionalitats de l'eina

Cadascuna de les pestanyes que apareixen a la interfície ens permet realitzar una consulta o prova diferents, en els següents sub-apartats es descriuen les diferents opcions permeses per l'eina.

5.2.1 Interfaces

Aquesta primera pestanya, ens permet consultar dades emmagatzemades als serveis d'arxius de mesures.

Mitjançant l'opció *Select Service Adresses*, situada al panell esquerre, podem seleccionar el servidor de mesures a consultar, després podem utilitzar la opció *Load Metadata* que ens carregarà la informació de les interfícies definides al servei. També podem acotar la cerca indicant la data d'inici i la duració de les dades que volem obtenir.

Aquesta pestanya doncs, permet la connexió a serveis *SQL-MA* i *RRD-MA* que hagin estat definits per perfSONAR al arxiu de serveis disponibles o bé que nosaltres definim directament amb un arxiu de configuració.

Per a utilitzar, per exemple, el servei RRD-MA que s'ha instal·lat, des de l'opció *Select Service Address*, podem modificar un dels registres que s'indiquen, posant la URL d'entrada al nostre servei, i definint els tipus d'events que accepta el servidor i l'esquema utilitzat per a contactar-hi.

A l'annex M s'inclouen instruccions per a la creació i càrrega d'arxius que faciliten la consulta d'aquells serveis que s'utilitzarà amb més freqüència.

5.2.2 *Circuits*

En aquesta ocasió observem una pestanya d'un servei que no s'inclou a la distribució MDM de perfSONAR, el servei HADES MA [34] (*Hades Active Delay Evaluation System*). Aquest servei permet la mesura del retard entre els extrems d'un enllaç determinat i de forma bidireccional. La direcció vindrà marcada depenent de quin extrem sigui el que realitzi la prova.

5.2.2.1 *El servei HADES*

Aquest realitza mesures de retard a través de punts dedicats de la xarxa, dels quals s'emmagatzemen les dades en un servidor i es poden consultar els detalls a través de la perfSONAR UI.

Aquest tipus de proves es realitzen a partir de la tramesa periòdica de grups de paquets UDP d'un punt de mesura a un altre cada 30 o 60 segons. El punt receptor desa el valor de les marques temporals de cada paquet en un arxiu de text que es genera diàriament. Per cada paquet es desen el nombre de seqüència i la marca temporal d'emissió i recepció.

Posteriorment aquestes dades son processades línia a línia, on cadascuna es correspon a un paquet, per a posteriorment extreure la informació més rellevant.

Un cop seleccionat el servidor que volem consultar, a través de l'opció *Select service addresses* i fent clic a *Retrieve all*, apareix una matriu amb tots els enllaços per als que es recullen dades. Cal doncs seleccionar un d'aquests enllaços entre diferents organismes adscrits a Géant2 per a rebre la informació de les mesures que proporciona el servei.

El servei ens mostra les següents mesures:

- Variació del retard de *jitter* IP
- *Jitter* màxim de paquet IP
- *Jitter* mínim de paquet IP
- Retard màxim
- Retard mínim

- Mitjana del retard
- Nombre de paquets duplicats
- Nombre de pèrdues

A més, es permet filtrar els resultats obtinguts segons alguns dels valors de les mesures, com ara el la mida del grup de paquets UDP enviats o la mida dels paquets.

5.2.3 **BWCTL**

Aquesta pestanya permet la realització de tests d'ample de banda sota demanda utilitzant el servei BWCTL comentat anteriorment a l'apartat 4.1.

Cal seleccionar els dos serveis BWCTL que volem que executin la prova d'ample de banda utilitzant les opcions *BWCTL MP Source* i *BWCTL MP Destination* i podent seleccionar també determinades opcions de la connexió com són:

- Protocol de transport (TCP o UDP)
- Duració de la prova
- Interval de report
- Mida de la finestra de TCP
- *Round Trip delay Time*

La opció *Estimate RTT* ens permet realitzar una estimació del temps d'anada i tornada en mil·lisegons. També permet indicar un valor d'RTT i la mida de la finestra de transmissió TCP, però la pròpia interfície indica que aquests valors es determinen automàticament pel servei BWCTL, i que les entrades manuals son tan sols per a proves específiques.

Un cop s'executa el test, els servei ens retorna els valors d'ample de banda assolits en cada interval de report, i el total. La interfície també mostra els valors a una gràfica.

5.2.4 **Looking Glass**

La pestanya *Looking Glass* és la finestra des de la que podem fer un cop d'ull a una xarxa a través del servei *Telnet/SSH* descrit al punt 4.2.

Trobem durant la realització d'aquesta memòria que els servidors que implementen aquest servei a través de la perfSONAR UI o bé no son disponibles (*Status* en vermell) o bé presenten errors de connexió (*Status* en verd) tal com es pot veure a la imatge 5.2 a continuació.

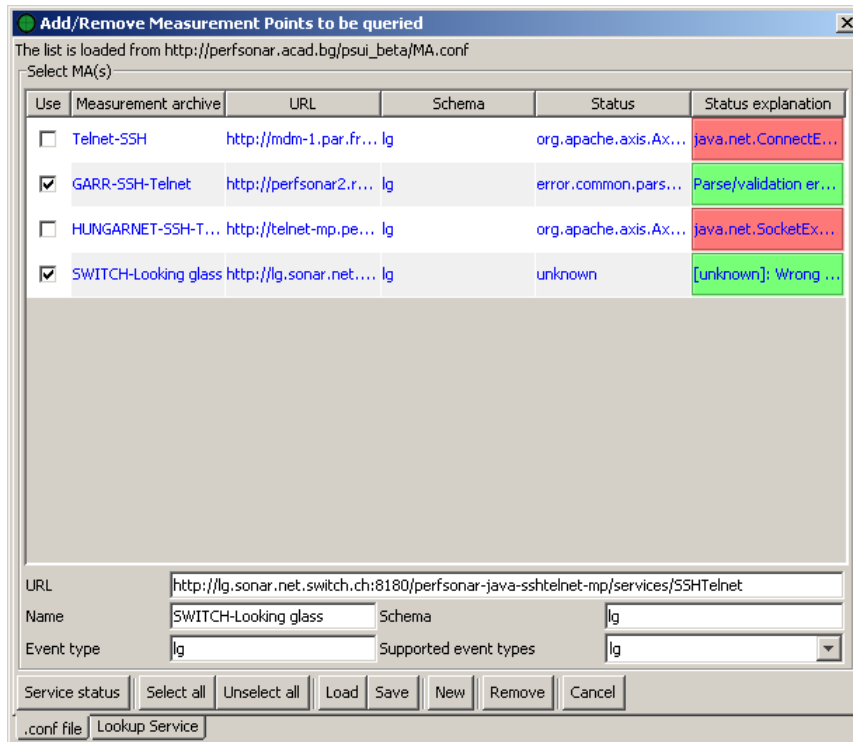


Fig. 5.2 Pantalla de configuració dels serveis a consultar

A la part esquerra de la pantalla principal es pot seleccionar el dispositiu que es vol consultar i el punt de mesura a consultar, rebent com resposta una llista dels dispositius de xarxa configurats al servei.

Un cop indicat el dispositiu que volem observar, a la part dreta es poden seleccionar les comandes que han estat configurades per aquest dispositiu. Un cop finalitzat s'executa la consulta a través de la opció *Query Data* que ens indicarà la resposta al quadre inferior *Response*. Donat que no hi ha disponibilitat de serveis, no es poden mostrar proves realitzades amb aquest entorn

5.2.5 FlowSA

Aquesta pestanya permet consultar serveis de mesures *Flow Selection and Aggregation*. Aquests serveis s'encarreguen de l'emmagatzematge de dades generades per l'eina *nfdump* [35], que captura el tràfic des de un dispositiu de xarxa i passat un temps 'x' envia les dades al servei 'central'.

Així doncs el FlowSA s'encarrega de proporcionar l'entrada a aquestes dades del servei, permetent als usuaris autoritzats la realització de consultes (en format

nfdump) als logs de l'aplicació, permetent així la recerca d'informació relacionada amb la seguretat o l'activitat de xarxa.

Donat que els dispositius de xarxa no tenen gaire capacitat per a desar gran quantitat de dades aquest servei pretén emmagatzemar tantes dades com permeti el servidor de l'aplicació.

De la mateixa manera que en el punt anterior, en aquests moments no es troba cap servidor disponible per a la consulta a un servei FlowSA.

Per a realitzar una consulta, cal indicar un servei disponible, i a l'apartat *Options* definir el tipus d'informació que volem rebre, podent escollir entre diferents paràmetres com el tipus de registres a mostrar, els dispositius de xarxa disponibles, el període de principi i fi, etc.

5.2.6 Playground

Aquesta pestanya és la interfície de proves per a serveis de perfSONAR a través de la perfSONAR UI, és a dir, a través d'ella podem enviar sentències XML de les que esperem que el servei que volem provar, reaccioni.

La pestanya inclou determinades consultes ja definides per a provar la connexió a un servei que acabi de ser instal·lat, o per a enviar una consulta qualsevol definida per l'usuari.

Durant la realització d'aquest treball la pestanya *playground* s'ha utilitzat per a la comprovació del funcionament dels diferents serveis i també va ser útil per a trobar la manera que els serveis de mesura d'arxius poden rebre les mesures per a desar-les en bases de dades, com es comenta al punt 3.1.7.

L'annex D mostra com s'utilitza la pestanya Playground per provar el correcte funcionament d'un servei instal·lat.

5.2.7 Lookup Service Playground

Igual que a l'apartat anterior, ens trobem amb una pestanya de proves, però en aquest punt es tracta d'una interfície destinada a proves a realitzar contra serveis de Lookup.

5.3 Desenvolupament de noves funcionalitats

Com a segona part de recerca d'aquest treball, es requereix la implementació de noves funcionalitats per la interfície gràfica de perfSONAR. En aquest apartat es documenten el codi necessari per a afegir noves pestanyes per a la interfície de perfSONAR amb l'ànim de permetre la futura integració de noves aplicacions a aquest entorn.

5.3.1 Preparació de l'entorn de desenvolupament

Com s'ha comentat anteriorment, l'entorn perfSONAR està desenvolupat utilitzant la tecnologia Java, i per tant es decideix utilitzar l'entorn de desenvolupament de codi obert Eclipse [36], que permet desenvolupar amb llenguatge Java així com altres llenguatges de programació com C, C++ i PHP entre d'altres, sempre que s'instal·lin els connectors corresponents.

El codi que es presenta en aquest treball ha estat compilat i executat utilitzant la màquina virtual de Java, JDK 1.5.0.22 [37].

Tal com s'indica a [38], perfSONAR permet accés de lectura a un servei SVN públic des del què es pot descarregar el codi que utilitzarem. La descarrega de codi es realitza mitjançant l'eina Subclipse, un *plugin* per a l'entorn de desenvolupament que ens permet desar el codi en un projecte local. Així doncs un cop tenim un projecte amb el codi que ens interessa, cal configurar l'entorn Eclipse indicant les llibreries externes de perfSONAR que es requereixen per a l'execució del codi..

Segons [39], el codi disponible al repositori de descàrrega no està actualitzat, però això no suposa un inconvenient per a la realització d'aquestes tasques donat que no han variat els objectes clau que s'utilitzen per a incloure noves pestanyes a la interfície.

5.3.2 Afegint noves aplicacions

Un cop configurat l'entorn és el moment de començar a desenvolupar noves funcionalitats per a la interfície. Primer s'exposa el mínim que ha d'acomplir el codi per a integrar-se amb l'entorn perfSONAR.

5.3.2.1 Requisits

Segons s'observa a [40], cadascuna de les pestanyes consten d'un mínim de dues classes o arxius .java, el primer arxiu és el que s'encarrega de crear la pestanya a la perfSONAR UI, mentre que l'altre és on es defineix l'aspecte i

funcionalitats que realitzarà, podent intervenir més classes o arxius si s'hi afegeixen altres elements que ho requereixin com podrien ser gràfics.

- **IPerfsonarTab**

Cada classe tindrà el seu nom depenent del servei que ofereixi, però per aparèixer a la interfície gràfica, s'ha de crear una classe que hereti els mètodes i de la classe `PerfsonarModel`.

Aquesta classe ha de cridar al mètode heretat `createMainComponent` que s'encarregarà de dibuixar la nova pestanya a la interfície de `perfSONAR` els objectes que es defineixen a la classe que es comenta a continuació. Aquesta classe també ha de definir els mètodes `getOrder()` i `toString()` que defineixen l'ordre que ocuparà la pestanya a la finestra i el nom de la pestanya que estem creant.

A l'annex N es mostra una implementació mínima dels mètodes que cal implementar o sobreescrivre per a que la interfície carregui la pestanya que s'ha desenvolupat.

També cal definir les accions que durà a terme el codi. Aquestes accions estan totalment pensades per a l'entorn `perfSONAR` i donat que el nostre codi no realitza accions estàndard, s'han utilitzat les definicions d'accions de la pestanya `BWCTL` amb l'objectiu de que codi compili. Aquestes accions estan dintre de l'esquema de l'aplicació i les accions que s'han implementat en aquest TFC no contemplen la utilització de missatges o accions definides per als serveis de `perfSONAR`.

- **MainComponent**

Aquesta segona classe és la que conté els elements que hi haurà dins la pestanya que ha afegit la classe anterior. Cal destacar els mètodes `addWidget()`, que afegeix tots els elements que apareixen a la pestanya i el mètode `actionPerformed()`, que s'executa en el moment que l'usuari realitzi alguna acció o que indiqui al programa que cal realitzar les accions que s'han programat.

A l'annex N es pot veure una implementació d'un programa senzill del tipus "Hola Mon" integrat a la `perfSONAR UI`.

5.3.2.2 Pestanya MostraText

Per a aquesta primera pestanya, es demana una aplicació que pugui rebre text a través d'un formulari d'entrada i un cop s'executi l'acció, el text introduït es mostri a la mateixa pestanya.

Es presenten dues maneres de mostrar el text, utilitzant un camp de text tipus *JTextArea* i en un camp tipus *JLabel*. El codi d'aquesta aplicació es pot trobar a l'annex O.

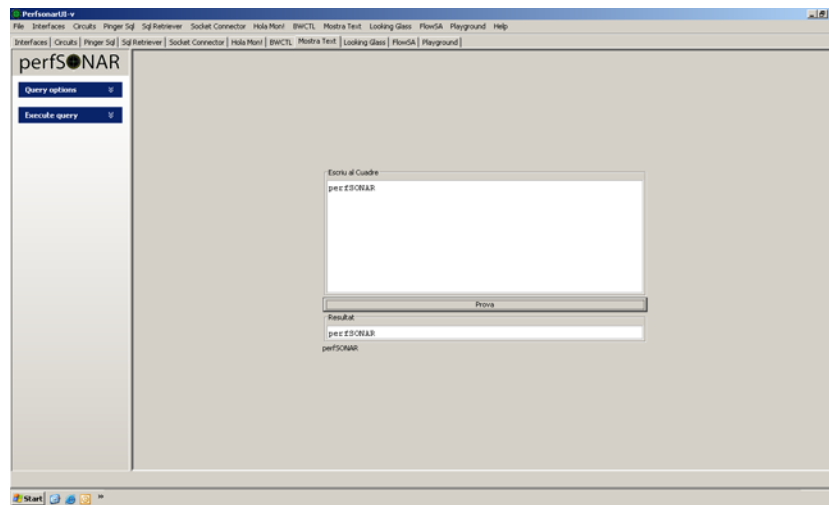


Fig. 5.3 Aspecte de la pestanya 'MostraText'

5.3.2.3 Pestanya Pinger Sql

La segona funcionalitat de mostra que s'ha realitzat consisteix en una classe que rep per paràmetre l'adreça IP d'un equip i crida l'eina ping del sistema per a enviar missatges *ping* al sistema que hi hem indicat.

Un cop arriben les respostes de l'equip remot, la interfície extrau el valor del RTT i connecta amb una base de dades mySQL, on s'emmagatzemen els valors obtinguts. Posteriorment l'eina mostra per pantalla el resultat del ping i els valors de RTT en una gràfica.

El codi d'aquesta interfície es pot consultar a l'annex P.

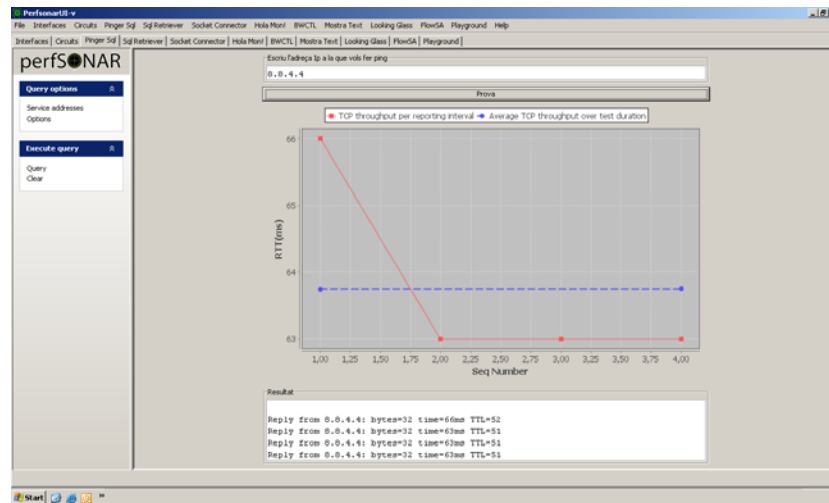


Fig. 5.4 Visualització de la pestanya PingerSql un cop efectuat la sèrie de pings

5.3.2.4 Classe Sql Retriever

La classe anterior emmagatzemava els valors de RTT a una base de dades SQL, doncs aquesta, permet la consulta d'aquestes dades. La classe rep mitjançant un formulari de text l'adreça IP del servidor que volem consultar. Un cop es té aquesta dada i s'executa l'acció, el programa connecta amb la base de dades mySQL i rep els valors de RTT emmagatzemats per al servidor que haguem sol·licitat i després els representa de forma textual i gràficament.

El codi d'aquesta pestanya s'ha adjuntat a l'annex Q.

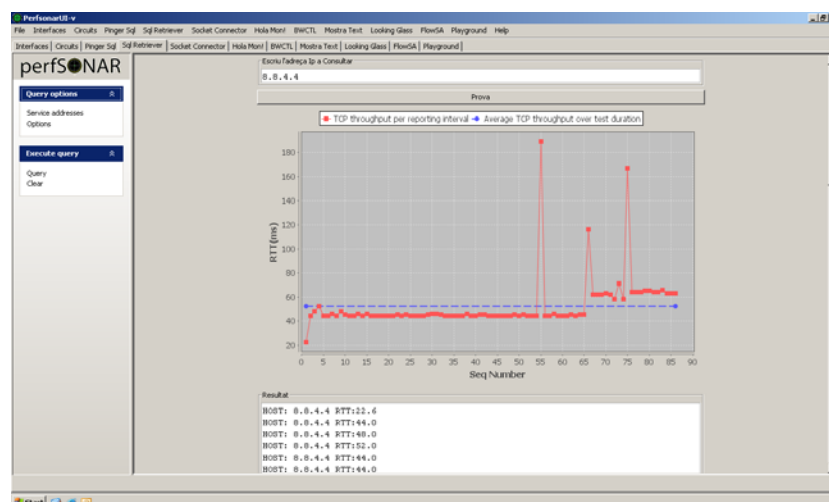


Fig. 5.5 Mostra de la pestanya Sql Retriever desenvolupada

5.3.2.5 Classe Socket Connector

La darrera pestanya realitzada es tracta d'una aplicació client, que es connecta a un servidor desenvolupat també per a mostrar el funcionament d'aquesta aplicació.

El servidor tan sols escolta peticions a un port determinat i quan rep una connexió espera a que l'aplicació client hi envii un text. Un cop el servidor rep aquest text, ha de contestar amb la mateixa cadena de caràcters, i respondre amb el mateix text que ha enviat el client per pantalla. Així doncs la connexió establerta, en aquest cas TCP, es realitza mitjançant les llibreries de xarxa proporcionades per Java.

D'aquesta manera, juntament amb les altres aplicacions proposades, podem determinar que és possible establir connexions des de la consola perfSONAR UI, que ens permetrien modificar el comportament d'un altre servei adequadament programat per a rebre aquest tipus de peticions.

El codi de les aplicacions client-servidor es pot consultar a l'annex R

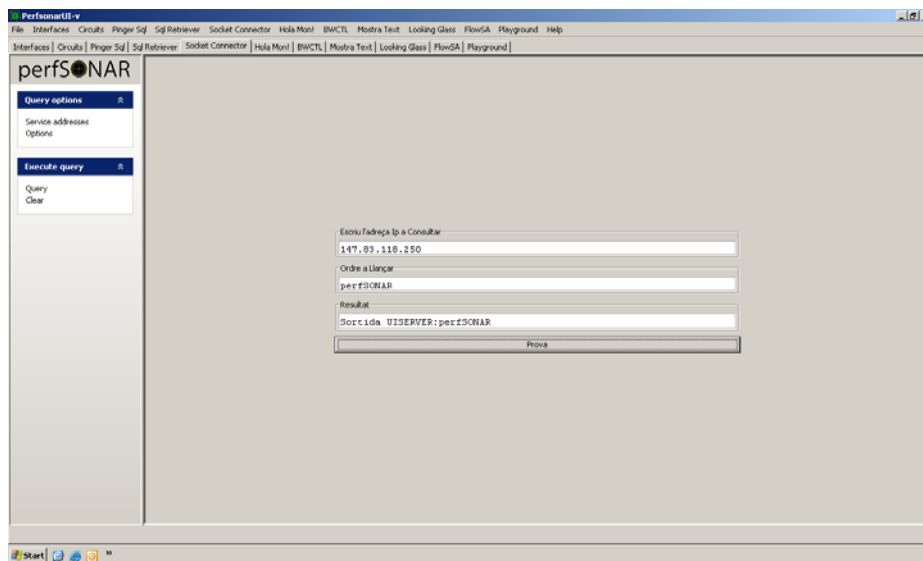


Fig. 5.6 Pantalla de configuració dels serveis a consultar

CAPITOL 6. ALTRES ACTIVITATS, LÍNIES FUTURES I CONCLUSIONS

Aquest darrer capítol comenta altres tasques realitzades que tenen relació però no són dintre de l'àmbit d'estudi, per a continuar amb unes línies sobre el impacte mediambiental, línies futures de treball que s'obren a partir d'aquest projecte i finalment les conclusions.

6.1 Altres tasques realitzades

Durant la realització d'aquest treball, els programadors de l'eina NDT, que permet realitzar un diagnòstic de problemes de connexió a través de proves de xarxa, van demanar a la llista de correu de perfSONAR voluntaris per a la traducció dels missatges que es mostren a l'eina. De manera que vam oferir la traducció dels missatges al Català i al Castellà. La traducció es pot consultar a l'annex S. Una membre del Centre de Supercomputació de Catalunya es va posar en contacte amb l'autor i el director d'aquest TFC mostrant interès per utilitzar els arxius en Català.

A més, també s'ha participat a les llistes de correu, aconsellant a d'altres usuaris que començaven a instal·lar els serveis de perfSONAR.

6.2 Estudi d'ambientalització

Les xarxes cada cop son més imprescindibles per a les comunicacions, de manera que, una correcta gestió de dispositius i enllaços podria permetre un estalvi considerable d'energia.

Les eines que proporciona el sistema perfSONAR, o fins i tot eines que encara s'hagin de desenvolupar, poden ajudar considerablement a la realització d'estudis d'enginyeria de tràfic que permetria a l'hora la presa de determinades decisions, com l'aturada d'alguns enllaços, servidors i/o routers, que per tant minimitzaria el consum elèctric derivat de les comunicacions.

6.3 Línies futures

Aquest treball aporta un anàlisi i una documentació adequada per a que l'Escola pugui participar en el projecte de la xarxa de recerca i d'educació europea Géant 3.

En d'altres treballs es podria implementar un dels serveis perfSONAR que no s'inclou a la distribució MDM, el FLOWSA (que permet accedir a dades de monitorització de fluxos IP), en un escenari de xarxa que simula l'esquema de RedIris implementada amb routers de menor capacitat però que utilitzen VLANs i taules d'encaminament reals. Aquest escenari s'ha realitzat al laboratori C4-325.

Una altra possibilitat seria l'estudi de matrius de trànsit. Les matrius de trànsit s'obtenen a partir de la mesura del tràfic entre nodes. L'anàlisi d'aquestes matrius pot donar indicacions sobre el comportament de la xarxa no a nivell local, sinó a nivell global, permetent una millora en la gestió i optimització dels recursos de xarxa. L'esquema de xarxa que s'indica anteriorment pot ser un bon entorn de proves per a la realització de programes que realitzin aquest anàlisi. Si finalment la recerca en aquest àmbit de les matrius de trànsit permet la creació d'eines d'anàlisi per grans xarxes, seria un servei molt indicat per a la inclusió a l'entorn perfSONAR, per exemple amb un mòdul addicional que facilités als operadors l'anàlisi del trànsit.

També es pot valorar la possibilitat d'implementar aquests serveis per a la monitorització dels enllaços de l'EETAC i proporcionar també punts de mesura per a realitzar proves entre xarxes.

6.4 Conclusions

L'objectiu principal d'aquest treball queda molt sintetitzat en el títol, 'Desplegament anàlisi i millora d'un sistema distribuït de mesures en xarxes de nova generació'. Es demanava la instal·lació, documentació i anàlisi del sistema de monitorització d'entorns multi-domini perfSONAR a més de l'anàlisi del funcionament de l'eina perfSONAR UI i el desenvolupament de codi d'exemple que pugui permetre la integració de noves eines en un futur.

L'entorn perfSONAR es perfila com un sistema de monitorització molt adequat per a un entorn multi-domini com és el de les xarxes de recerca. Un clar escenari d'aquestes característiques es dona a Géant2 on hi participen les xarxes nacionals de recerca dels països adscrits, i s'hi estableixen una sèrie de fronteres lògiques que el sistema perfSONAR pretén fer desaparèixer en quant a l'administració dels recursos de la xarxa de recerca i d'educació.

Tenint en compte que perfSONAR està encara en desenvolupament, tot apunta a que en un futur i si s'estableixen més acords de recerca entre centres, més institucions utilitzaran aquest tipus de sistema per a assegurar un correcte funcionament de les comunicacions.

En quant a la instal·lació dels serveis de perfSONAR, la distribució escollida, perfSONAR MDM, no es la més còmoda per a implementar. Solucions com el

pS-Performance Toolkit (Live Cd) o un DVD amb una pre-instal·lació de la versió anterior del MDM realitzat per RedIRIS, semblen a priori més senzilles d'utilitzar, ja que inclouen tots els paquets necessaris i part de la configuració que es realitza de forma assistida. En tot cas, no trobem una errada la distribució escollida, ja que ens sembla que és la millor manera de conèixer en profunditat les eines que conformen perfSONAR i com es produeix el diàleg entre serveis locals i globals, una necessitat vital per a un entorn distribuït.

S'han instal·lat correctament tots els serveis de la distribució perfSONAR MDM en la seva versió 3.2, dels quals s'ha pogut configurar i realitzar proves amb els serveis '*Lookup Service*', '*RRD-MA*' i '*BWCTL-MP*'. Es donen també indicacions sobre el funcionament dels components individuals.

Per altra banda, s'ha documentat el codi necessari per a la creació de noves funcionalitats de la interfície perfSONAR UI, i es proporcionen també codis d'exemple.

L'autor valora molt positivament l'experiència personal d'haver participat en una petita part d'un projecte d'àmbit Europeu, i la relació amb d'altres usuaris i experts d'altres centres. A més ha proporcionat a l'autor més fluïdesa en la utilització d'entorns Linux i més pràctica amb la programació de codi Java.

BIBLIOGRAFIA

- [1] Consorci perfSONAR, “perfSONAR a PERFORMANCE Service Oriented Network monitoring Architecture”, www.perfsonar.net (2006).
- [2] Comissió Europea, Xarxes nacionals d'educació i recerca, DANTE, “Géant2, Europe's research and education community network”, www.geant2.net (2004).
- [3] Ministeri d'indústria turisme i comerç, Ministeri de ciència i innovació, Red.es., “Rediris Red española para Interconexión de los Recursos InformáticoS”, www.rediris.es (1988).
- [3] Consorci universitats americanes, “Internet2 research and education communities networking consortium”, www.internet2.edu (1996).
- [4] DOE Office of Science, “Energy Sciences Network”, www.es.net (1987).
- [5] Consorci perfSONAR, “perfSONAR production release – downloads page”, <http://www.perfsonar.net/download.html> (2006).
- [6] Howlet, J., Nordh, V, Singer, W., “Deriverable DS3.3.1: eduGAIN service definition and policy Initial Draft”, http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-10-081-DS3_3_1_eduGAIN_service_definition_and_policy.pdf (2010).
- [7] Oetiker, T., “RRDTool, High performance data logging and graphing system for time series data”, <http://www.mrtg.org/rrdtool/> (1999).
- [8] Widenius, M., Axmark, D., Oracle, “MySQL The world's most popular open source database”, www.mysql.com (1994).
- [9] Delvaux, A., Kramer, G., Pikusa, P., Reale, M., Trocha, S., Vicinanza, D., “perfSONAR MDM 3.2 Administrator's guide”, http://www.perfsonar.net/download/bundles/perfsonar-mdm-bundle/3.2/perfSONAR_MDM_3.2_Admin_Guide_1.2.pdf (2009).
- [10] Col·laboradors Wikipedia, “Debian”, <http://es.wikipedia.org/wiki/Debian> (data indeterminada).
- [11] Debian Release Managers, “Informació ‘Debian 6.0 “Squeeze” frozen””, <http://www.debian.org/News/2010/20100806> (2010).
- [12] Sun Microsystems - Oracle, “Conozca más sobre la tecnología Java”, <http://www.java.com/es/about/> (1996-2010).

- [13] Apache Foundation, "Apache Tomcat an Open Source JSP and Servlet Container", <http://tomcat.apache.org/> (2001-2010).
- [14] Meier, W., "eXist, an Open Source repository and retrieval engine for XML documents", <http://exist.sourceforge.net/index.html> (2000).
- [15] Rodriguez, C., "Authentication Service – JRA 1 Meeting, Utrecht 28th & 29th March 07", [https://wiki.man.poznan.pl/perfsonar-mdm/images/perfsonar-mdm/3/34/PerfSONAR - AS - 070328.pdf](https://wiki.man.poznan.pl/perfsonar-mdm/images/perfsonar-mdm/3/34/PerfSONAR_-_AS_-_070328.pdf) (2007).
- [16] Rodriguez, C., Comunicació personal (e-mail), Novembre 2009.
- [17] Llista de correu perfsonar-user, Comunicació personal (e-mail), Abril 2010
- [18] Paul, G., "IPTraf a console-based network statistics utility for Linux", <http://iptraf.seul.org/> (1997).
- [19] Combs, G., "Wireshark", <http://www.wireshark.org/> (1998).
- [20] Begin, C., "iBATIS Data Mapper ", <http://ibatis.apache.org> (2002).
- [21] Begin, C, "myBatis Data Mapper", <http://www.mybatis.org/index.html> (2010).
- [22] Pikusa, P. Comuncació personal (e-mail) , Novembre 2009 – Octubre 2010.
- [23] Mills, D., "RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis", <http://tools.ietf.org/html/rfc1305> (1992).
- [24] pool.ntp.org Project, "Europe ntp pool zone", <http://www.pool.ntp.org/zone/europe> (2003).
- [25] Dugan, J., Kutzko, M., "Iperf a network testing tool", <http://sourceforge.net/projects/iperf> (2008).
- [26] Boote, J., Brown A., "Bandwith test Controller", <http://www.internet2.edu/performance/bwctl/index.html> (2009).
- [27] Calim, H., Cavara, S., Gründl, M., Comunicació personal (e-mail), Novembre 2009.

- [28] Consorci perfSONAR, “perfSONAR a PERformance Service Oriented Network monitoring Architecture”, www.perfsonar.net (2006).
- [29] Delvaux, A., Trocha, S., “perfSONAR Bundle Release Notes”, <https://wiki.man.poznan.pl/perfsonar-mdm/index.php/PerfSONAR-Bundle-Release-Notes> (2008-2010).
- [30] Zurawski, J., Comunicació personal (e-mail), Agost 2010.
- [31] Escolano, A., “Presentación perfSONAR – Salamanca 15 de Junio de 2010”, <http://www.rediris.es/qt/qt2010/ponencias/PerfSONAR-RedIRIS.aes.ppt>. (2010)
- [32] Leninen, S., “End-System (Host) Tunning”, <http://kb.pert.geant.net/PERTKB/EndSystemTuning> (2009).
- [33] Sun Microsystems - Oracle, “¿Qué es Java Web Start y cómo se ejecuta?”, http://www.java.com/es/download/faq/java_webstart.xml (data indeterminada).
- [34] Kraft, S., “Hades MA Service”, https://wiki.man.poznan.pl/perfsonar-mdm/index.php/Hades_MA_Service (2008).
- [35] Haag, P., Jändling, T., “NFDUMP – Netflow processing tools”, <http://nfdump.sourceforge.net/> (2006).
- [36] The Eclipse Foundation, “Eclipse an Open Source universal toolset for development”, <http://www.eclipse.org/> (2001-2010).
- [37] Sun Microsystems - Oracle, “Java JDK 1.5.0.22”, <http://www.oracle.com/technetwork/java/javase/downloads/index-jdk5-jsp-142662.html> (2001-2010).
- [38] Molenaar, G., Rodriguez, C., Trocha, S., “perfSONAR Developers Information”, https://wiki.man.poznan.pl/perfsonar-mdm/index.php/PerfSONAR_Developers_information (2008-2010).
- [39] Pikusa, P., Comunicació personal (e-mail), Agost – Setembre 2010
- [40] Jeliazkova, N., “perfSONAR’s source code documentation – Javadoc for perfONARUI v0.10”, <http://perfsonar.acad.bg/doc/> (2005-2007).

GLOSSARI D'ABREVIATURES UTILITZADES EN AQUEST DOCUMENT

AS:	Authentication Service
BWCTL:	BandWith ConTrolLer
ESNET:	Energy Sciences NETwork
FLOW SA:	Flow Selection and Aggregation
gLS:	global Lookup Service
HADES MA:	Hades Active Delay Evaluation System
hLS:	home Lookup Service
HTML:	HyperText Markup Language
HTTP:	HyperText Transfer Protocol
JDK:	Java Development Kit
JRE:	Java Runtime Enviroment
JSP:	Java Server Pages
LS:	Lookup Service
MDM:	Multi-Domain Monitoring
NREN:	National Reseach and Education Network
NTP:	Network Time Protocol
perfSONAR:	PERFormance Service Oriented Network monitoring Architecture
PHP:	Php Hypertext Pre-processor o Personal Homepage Tools
RedIRIS:	Red Española para la Interconexión de los Recursos InformáticoS
RNP:	Rede Nacional de Ensino e Pesquisa
RRD:	Round Robin Database
RRD-MA:	Round Robin Database Measurement Archive
RTT:	Round Trip delay Time
SNMP:	Simple Network Management Protocol
SOAP:	Simple Object Access Protocol
SQL:	Structured Query Language
SQL-MA:	Structured Query Language Measurement Archive

SSH:	Secure SHell
TCP:	Transmission Control Protocol
UDP:	User Datagram Protocol
UI:	User Interface
XHTML:	eXtensible HyperText Markup Language
XML:	eXtensible Markup Language.

ANNEX A. ARXIUS DE CONFIGURACIÓ DEL SERVEI APTITUDE

S'inclouen en aquest annex els arxius de configuració del servei aptitude encarregat de la instal·lació dels paquets disponibles per a la seva instal·lació als repositoris que estiguin configurats a l'arxiu `/etc/apt/sources.list`

El signe '#' indica que la línia és un comentari.

#

#La primera font de descàrrega (no implica prioritat) és el #propi CD d'instal·lació

deb cdrom:[Debian GNU/Linux 5.0.3 _Lenny_ - Official amd64 CD Binary-1 20090905-11:02]/ lenny main

#Repositori de rediris, s'instal·la aquest repositori de forma #automàtica per localització geogràfica

deb http://ftp.rediris.es/debian/ lenny main

deb-src http://ftp.rediris.es/debian/ lenny main

deb http://ftp.rediris.es/debian etch main contrib non-free

#Repositori de seguretat de Debian, on es publiquen les #actualitzacions crítiques de vulnerabilitats que poden #comprometre la seguretat del sistema

deb http://security.debian.org/ lenny/updates main

deb-src http://security.debian.org/ lenny/updates main

#Aquest repositori conté canvis necessaris per a mantindre el #funcionament de programes estables, com filtrat de Spam o #antivirus

deb http://volatile.debian.org/debian-volatile lenny/volatile main

deb-src http://volatile.debian.org/debian-volatile lenny/volatile main

#Repositori local per a la instal·lació del paquet librrd2 no #disponible als repositoris de Debian.

deb http://localhost/repositori/apt/ lenny main

#Repositori estable dels serveis de perfSONARdeb
http://downloads.perfsonar.eu/repositories/deb/ main

ANNEX B. INSTAL·LACIÓ I CONFIGURACIÓ DELS PREREQUISITS

B.1 Definició dels prerequisits de perfSONAR

Indiquem al procés del sistema '*apt*' nous repositoris a utilitzar.

```
# cd /etc/apt/sources.list.d
# wget
  http://downloads.perfsonar.eu/repositories/deb/perfsonar-
  stable.list
```

Descarreguem i importem la clau del paquet

```
# wget http://downloads.perfsonar.eu/repositories/perfsonar.asc
# apt-key list
```

Eliminem i actualitzem el repositori

```
# apt-get clean
# apt-get update
```

S'obre el fitxer de text `/etc/apt/sources.lst`, per a buscar la línia:

```
deb http://<Algun host> etch main contrib
```

I modificar-la com:

```
deb http://<Algun host> etch main contrib non-free
```

A l'annex A s'inclou l'arxiu de configuració del servei *aptitude* del perfHOST.

B.2 Java Virtual Machine

B.2.1 Instal·lació

En el nostre cas, degut a haver d'instal·lar el servei Apache Tomcat, per a la instal·lació de serveis perfSONAR, es requereix la instal·lació de la màquina virtual de desenvolupament en la versió 5. La instal·lació es fa a través de la consola de comandes com a usuari root, utilitzant la següent comanda:

- `Apt-get install sun-java5-jdk`

Un cop acceptat el missatge de confirmació, el sistema descarregarà els paquets necessaris del repositori de Debian i els instal·larà.

B.2.2 Configuració màquina virtual de Java

Un cop instal·lada la màquina virtual recomanada per perfSONAR, s'ha d'indicar al sistema operatiu quina de les màquines virtuals disponibles ha utilitzar, ja que Debian incorpora una màquina virtual preinstal·lada.

Per a indicar al sistema operatiu quina màquina virtual utilitzarà, hem de fer-ho com a usuari root a través de la consola de comandes, utilitzant la següent sentència:

- `update-alternatives --config java`

Ens apareixerà una pantalla en la que s'ha d'indicar d'entre les opcions la versió 5 oficial de Sun Microsystems.

De la mateixa manera li podem indicar al sistema quin és el motor de compilació per a codi java (si en tenim més d'un). Igual que anteriorment, es fa a partir de la consola de comandes, variant lleugerament la sentència anterior.

- `Update-alternatives --config javac`

En el cas del perfhost, només tenim instal·lat un motor, per tant, no el podem canviar.

B.3 Apache Tomcat

B.3.1 Instal·lació

Per a la instal·lació del Tomcat, un cop configurada la màquina virtual de Java , s'ha d'executar la següent comanda amb privilegis de root:

```
# apt-get install tomcat5.5-exist
```

En aquest punt s'està instal·lant una distribució específica de perfSONAR que conté la base de dades XML eXist, que s'explica a l'annex C

Un cop finalitzada la instal·lació el sistema arrenca el servei de forma automàtica.

Es verifica el correcte funcionament del servei accedint a la següent adreça en el nostre navegador web.

<http://ipdelservidor:8180>

en el nostre cas, la URL resultant és:

<http://147.83.118.250:8180>

B.3.2 Modificacions de la configuració

Per al correcte funcionament dels serveis de perfSONAR, s'ha de realitzar una modificació en els arxius de configuració. Aquesta modificació s'ha de fer per a permetre que l'aplicació pugui escriure determinats arxius al disc, ja siguin arxius de configuració o bé arxius de log que continguin informació dels accessos o el tipus d'accessos que hi ha hagut al servei.

Per això s'ha de modificar l'arxiu de configuració:

- `/etc/default/tomcat5.5`

Modificant el valor del paràmetre TOMCAT5_SECURITY a "no".

S'inclou també a continuació una mostra de l'arxiu de configuració resultant per al funcionament dels serveis de perfSONAR, segons es recomana a la guia d'administració dels serveis.

```
# Run Tomcat as this user ID. Not setting this or leaving it #blank
will use the default of tomcat55.
#TOMCAT5_USER=tomcat55
# The home directory of the Java development kit (JDK). You need at
least
# JDK version 1.4. If JAVA_HOME is not set, some common #directories
for the Sun JDK, various J2SE 1.4 versions, and the #free runtimes
# java-gcj-compat-dev and kaffe are tried.
#JAVA_HOME=/usr/lib/jvm/java-6-sun
# Directory for per-instance configuration files and webapps. It
contain the
# directories conf, logs, webapps, work and temp. See RUNNING.txt for
details.
# Default: /var/lib/tomcat5.5
#CATALINA_BASE=/var/lib/tomcat5.5

# Arguments to pass to the Java virtual machine (JVM).
#JAVA_OPTS="-Djava.awt.headless=true -Xmx128M"
```

```
# Java compiler to use for translating JavaServer Pages (JSPs). You can
use all
# compilers that are accepted by Ant's build.compiler property.
#JSP_COMPILER=jikes
# Use the Java security manager? (yes/no, default: yes)
# WARNING: Do not disable the security manager unless you understand
# the consequences!
# NOTE: java-gcj-compat-dev currently doesn't support a security
# manager.
TOMCAT5_SECURITY=no
CATALINA_OPTS="-Djava.awt.headless=true -Xmx128M -server -
Djava.library.path=/usr/lib"
```

Després de realitzar alguna modificació acció sobre la configuració de Tomcat, és molt recomanable realitzar un reinici del servei. Per a realitzar un reinici, parada o arrancada del servei Tomcat s'ha de fer utilitzant la sentència que correspongui:

```
# /etc/init.d/tomcat (restart || stop || start)
```

B.2.1 Configuració del servei de Lookup local a la resta de serveis de perfSONAR

Els arxius de mesures i els punts de mesura de l'entorn perfSONAR permeten la configuració del registre envers un Lookup Service que permet doncs l'anunci dels serveis disponibles a la resta del sistema global de perfSONAR.

Acostuma a aparèixer la opció 'Do you wish to register with an LS?' a les planes de configuració dels seus serveis, i un cop es selecciona 'yes' apareixen les següents opcions:

- *Enter the service name*
Requereix que s'indiqui el nom del servei que s'ha instal·lat, aquest nom serà com estarà identificat el servei cap a la resta del sistema perfSONAR.
- *Enter a description for the service*
Es pot afegir una petita descripció del servei que s'ha instal·lat
- *Enter the service administrator's email address*
Es pot especificar l'adreça de contacte de l'administrador del servei per si es produeix qualsevol tipus de problema.
- *Enter the name of the organisation running this service*
El sistema dona també informació de la organització que proporciona aquest servei.

- *Enter the LS URL*

Aquí definim amb quin servei LS es vol registrar el servei, cal no confondre-la amb la URL de l'apartat de configuració, aquesta entrada apuntarà cap a 'localhost' sempre que el servei sigui instal·lat al mateix equip que el servei de lookup. En el nostre cas <http://localhost:8180/geant2-java-xml-ls/services/LookupService/>

- *Set the registration interval (milliseconds)*

Aquest valor estableix l'interval de temps entre missatges de registre, serveix per assegurar que si el servei està arrencat, el lookup service tindrà constància de la disponibilitat del mateix. El valor per defecte de 43200000 mil·lisegons (12 hores) és adequat.

ANNEX C. EL PANELL DE CONTROL WEB D'EXIST

C.1 El panell web d'eXist

A continuació veurem el panell de control del servei i les opcions que disposa. Per a això hem d'accedir al portal d'administració de la base de dades a la següent url:

<http://perfhos:8180/exist/admin/admin.xql>

En aquest punt s'han d'introduir l'usuari i la contrasenya per a accedir al servei de base de dades. En la instal·lació amb l'entorn perfSONAR, l'usuari i el password són respectivament '*admin*' i '*admin*'.

Un cop accedim, la pantalla principal ens mostra a mà esquerra les diferents opcions de les que disposem i a la part dreta tenim informació del funcionament del servei local, tal i com es pot veure a continuació a la figura 2.1.

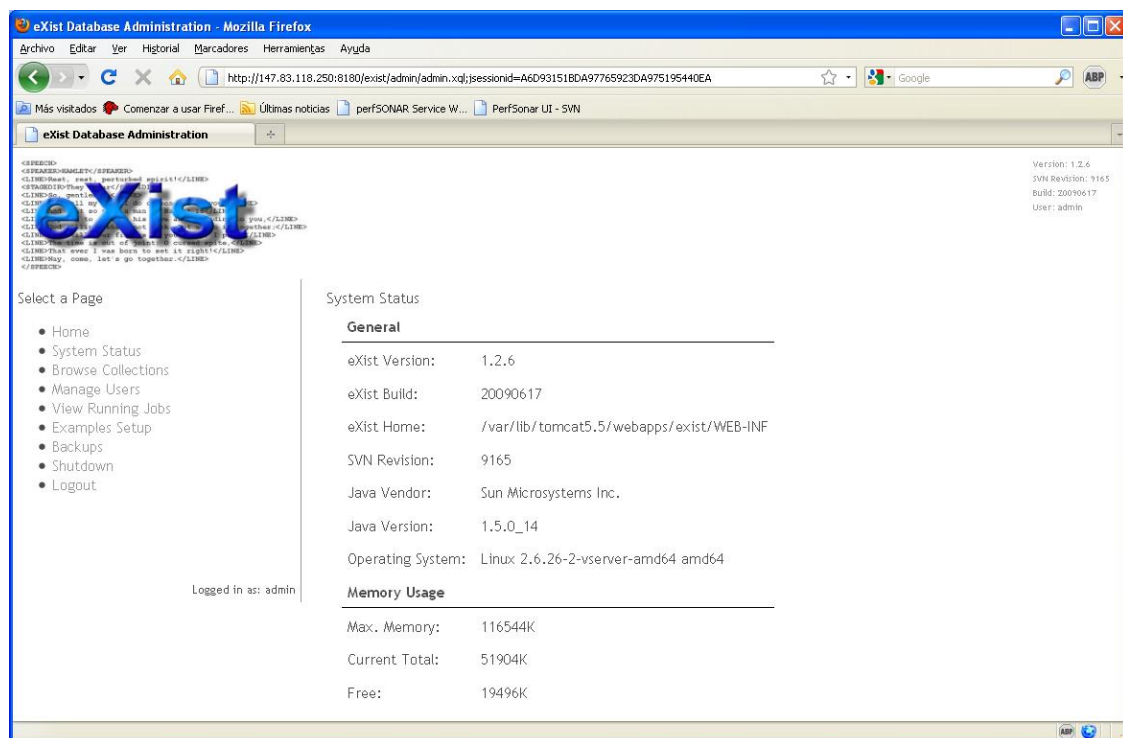


Fig 2.1 El panell de control del motor de base de dades XML eXist

A continuació es comenta que permet cadascuna de les opcions del menú:

- *Home*

Es correspon a la plana web de eXist que conté informació sobre el propi servei, des de aquí, podem accedir a diversos exemples per a mostrar el funcionament de la base de dades, així com enllaços a les planes web de desenvolupament, entre d'altres opcions.

- *System Status*

Com el seu nom en anglès ens indica, en aquesta pantalla es pot veure informació del servei instal·lat, com ara la versió instal·lada, el directori d'instal·lació, la màquina virtual de Java del sistema i el sistema operatiu. En aquest apartat l'eina també ens indica el nivell de memòria del servidor, indicant la memòria total, la memòria utilitzada i la memòria lliure.

- *Browse Collections*

En aquest apartat es llisten totes les bases de dades que hi hagi instal·lades en el sistema, de manera que cada línia es correspon a una base de dades. Si fem clic a alguna de les disponibles, el sistema mostrarà els arxius XML dels quals consta cada base de dades.

- *Manage Users*

Si seleccionem aquesta opció, el primer que trobarem són tots els usuaris que hi ha definits a l'entorn. Per usuari s'indica també el grup al qual pertany i l'arrel de cada usuari, és a dir a quina base de dades ha d'accedir, tot i que segons la configuració o el servei, pot aparèixer com a no establert (*not set*).

Des d'aquesta pantalla podrem editar els usuaris, canviar la contrasenya o la base de dades, crear-ne de nous o eliminar-los.

- *View Running Jobs*

També com el seu nom indica, ens permetrà veure quines tasques està executant en aquest moment el motor de base de dades.

- *Examples Setup*

Des d'aquesta opció, eXist ens permet carregar una sèrie de bases de dades d'exemple amb un contingut d'obres de Shakespeare, que s'utilitzaran per a mostrar el funcionament del motor en el punt C.2.

- *Backups*

Aquest apartat permetrà al usuari crear còpies de seguretat de totes les bases de dades que hi hagi instal·lades. Es crea un arxiu comprimit zip amb totes les bases de dades definides per l'usuari. Aquesta versió, però, no disposa d'una opció *Restore* que pugui recuperar l'entorn complet indicant on hi ha una còpia, segons [exist-db.org/backup.html] la versió 1.3 de eXist ja ho permet.

- *Shutdown*

Aquesta opció ens permet apagar el servei eXist, però al executar el servei des de Tomcat, aquesta acció també apagaria el servei Tomcat. En el formulari que apareix a la web es pot indicar el nombre de segons abans de que s'apagui.

- *Logout*

Permet acabar la sessió iniciada, si hi fem clic, el servei ens redirigirà a la plana d'entrada.

C.2 Funcionament del motor de bases de dades eXist

Per accedir i interactuar amb el servei eXist, s'utilitza el motor XQuery, que proporciona els mitjans per a l'extracció i manipulació d'arxius XML o qualsevol font de dades que es pugui representar amb aquest format.

Per a la revisió d'un arxiu XML, XQuery utilitza XPath, un llenguatge que permet la construcció d'expressions que processen un document XML per a seleccionar parts del contingut sense atributs. Xpath llegeix el contingut del document XML i crea un arbre segons les diferents etiquetes que es defineixin.

A més, XQuery incorpora també certes expressions similars al llenguatge SQL anomenades FLWOR ja que prenen el nom de les sentències *For*, *Let*, *Where*, *Order By* i *Return* pròpies de SQL. D'aquesta manera permet realitzar cerques més acurades utilitzant llenguatges de cerca àmpliament estesos en altres motors de bases de dades.

Les característiques de XQuery ens permeten realitzar accions com ara:

- Extracció de dades per a un servei web.
- Generació de resums de bases de dades XML.
- Realització de cerques textuais a la web.
- Selecció i transformació de dades XML a XHTML per a posterior publicació a una web.

- Obtenció de dades des de diferents fonts per a integrar-se a una aplicació.

Per mostrar el funcionament d'una consulta, ho farem a través de la interfície web del servei. Accedim a la plana principal del servei des d'un navegador web mitjançant l'adreça:

- <http://ipdelservidor:8180/exist>

A l'apartat *Examples* del menú situat a l'esquerra, seleccionem la opció *XQuery Sandbox*. Aquesta plana, tal i com es pot veure a la figura 2.2, conté una finestra per a fer la consulta amb els botons d'acció, a la part inferior, un altra finestra en la que es mostren els resultats de la consulta i a la part esquerra un apartat on podem desar mentre duri la sessió fins a deu consultes.

La primera consulta que farem a la base de dades consistirà en un simple *Hello World* que ens permetrà comprovar que el motor *XQuery* està correctament instal·lat.

Podem enganxar el següent codi al formulari de consulta:

```
1. xquery version "1.0";
2. let $message := 'Hello World!'
3. return
4. <results>
5.   <message>{$message}</message>
6. </results>
```

La línia 1 indica que s'està utilitzant la versió 1.0 del motor XQuery, la línia 2 defineix la variable message com a 'Hello World!'. La línia 3 conté la sentència return, que dona pas al format de la sortida que ve definit per les línies 4, 5 i 6. La línia 5 és la que indica, en aquest cas, que volem que es retorni el valor de la variable '\$message' inicialitzada a la línia 2.

Tal i com es pot veure a la figura 2.2 a continuació, el resultat que retorna el motor és:

```
1. <results>
2.   <message>Hello World!</message>
3. </results>
```

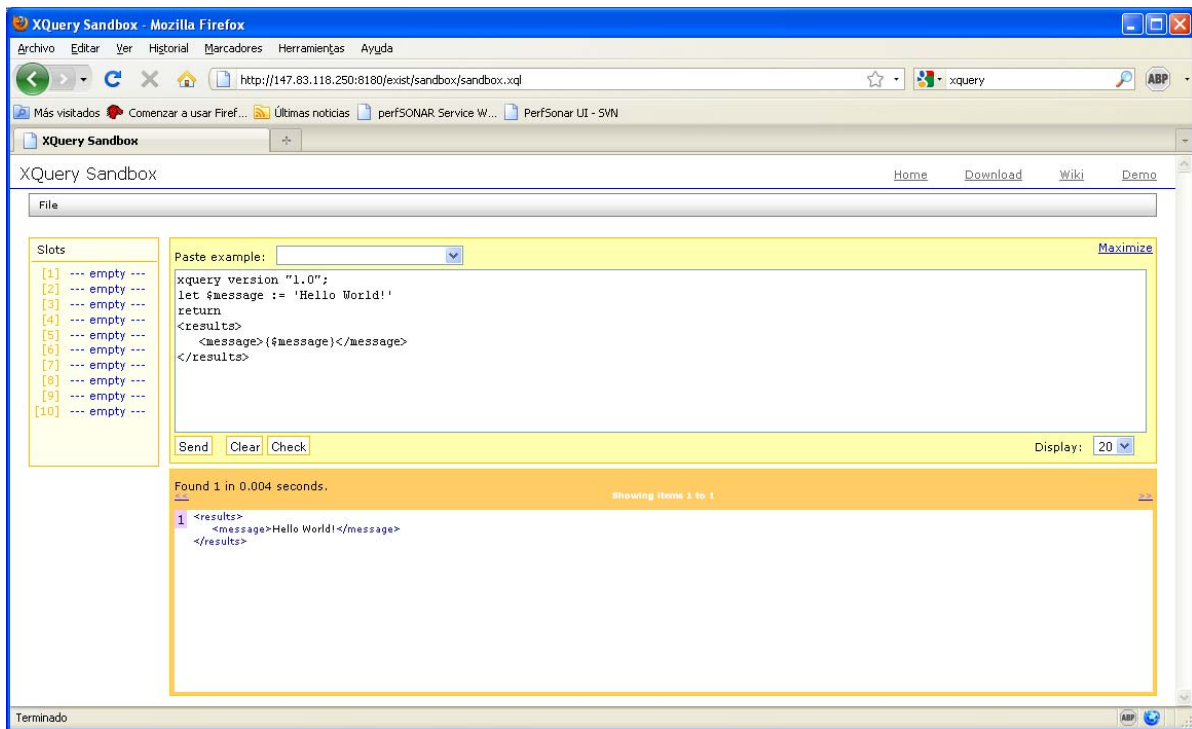


Fig 2.2 L'apartat XQuery Sandbox que ens permet realitzar proves amb el motor de bases de dades eXist.

Aquesta interfície també conté una sèrie de consultes predefinides una mica més complexes que l'anterior, en aquest cas, utilitzem una *query* que ens mostri cadascun dels personatges que apareguin a cada escena de cada acte de l'obra teatral Macbeth de William Shakespeare i que ens representará les dades de la següent manera:

```
<acte>
<Títol>Títol de l'Acte</Títol>
  <Escena>
    <Títol> Títol de l'escena </Títol>
    <actors>
      <actor>Nom Actor</actor>
      <actor>Nom Actor2</actor>
      ...
    </actors>
  </Escena>
</Escena>
...
</Escena>
</Acte>
```

Aquesta consulta es pot extreure utilitzant la següent sentència:

```
01. <toc>{
02.   for $act in doc("/db/shakespeare/plays/macbeth.xml")/PLAY/ACT
03.     return
04.       <act>
05.         {$act/TITLE}
06.         {
07.           for $scene in $act/SCENE return
08.             <scene>
09.               {$scene/TITLE}
10.               <actors>
11.                 {
12.                   for $speaker in distinct-values($scene//SPEAKER)
13.                     order by $speaker return
14.                       <actor>{$speaker}</actor>
15.                 }
16.               </actors>
17.             </scene>
18.           }
19.       </act>
20. }</toc>
```

D'aquesta manera, a la línia 1 i 20 indiquem l'inici i final de la taula de continguts resultant (*Table of Contents*), on volem cercar per tots els elements *act* i *play* de l'arxiu *macbeth.xml* definit a la línia 2. A la línia 3 s'indica la ordre '*return*', ja que a les línies 4 i 5 es defineix el retorn del nom del acte. A la línia 7 es defineix la cerca per cadascuna de les escenes, retornant els actors implicats amb la cerca definida entre les línies 9 i 16. Els diferents actors es seleccionen a la cerca de la línia 12, on s'estableix que per cada '*speaker*' diferent, es retorni la llista per ordre alfabètic. Les línies 16 a 19 corresponen al tancament de les cerques per actor, escena i acte.

ANNEX D. INSTAL·LACIÓ I CONFIGURACIÓ DEL *LOOKUP SERVICE*

D.1 Instal·lació del servei

Per a la instal·lació del servei, s'ha d'executar la següent comanda a la consola amb privilegis de root:

```
# apt-get install perfsonar-java-xml-ls
```

Un cop finalitzada la operació, el sistema reiniciarà automàticament el servei Tomcat.

Cal ara realitzar una còpia de les llibreries tipus '*jar*' del servei eXist al directori on es desen les llibreries que ha d'utilitzar el servei tomcat.

```
# cp /var/lib/tomcat5/webapps/exist/WEB-INF/lib/resolver-1.2.jar  
/var/lib/tomcat5/common/endorsed  
  
# cp /var/lib/tomcat5/webapps/exist/WEB-INF/lib/xalan-2.7.1.jar  
/var/lib/tomcat5/common/endorsed  
  
# cp /var/lib/tomcat5/webapps/exist/WEB-INF/lib/xml-apis.jar  
/var/lib/tomcat5/common/endorsed  
  
# cp /var/lib/tomcat5/webapps/exist/WEB-INF/lib/serializer-2.9.1.jar  
/var/lib/tomcat5/common/endorsed  
  
# cp /var/lib/tomcat5/webapps/geant2-java-xml-ls/WEB-  
INF/lib/xercesImpl-2.8.0.jar /var/lib/tomcat5/common/endorsed
```

Per a la instal·lació de la resta de components, s'assumeix que el servei de *Lookup* ha estat instal·lat i per tant aquestes llibreries ja es troben en aquest directori.

D.2 Comprovació del servei

Per a comprovar el funcionament del servei, hem d'accedir a través d'un navegador web a la següent adreça:

- <http://perfhost:8180/geant2-java-xml-ls/>

Al accedir, podem utilitzar la opció Test Service. Un cop es premi el botó 'test' la prova del sistema verificarà que es pugui contactar amb l'aplicació i ens informará del resultat, tal i com es pot veure en la figura 2.6 a continuació.

Deployment test

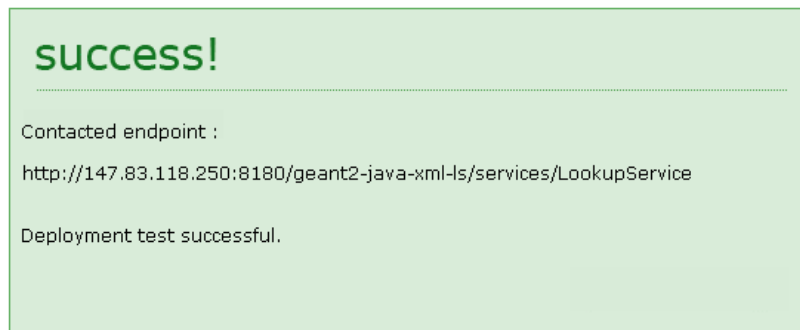


Fig. D.1 Captura de pantalla del resultat de la prova d'instal·lació del servei integrat a la pròpia aplicació

Per a realitzar una prova del funcionament del servei que s'acaba d'instal·lar amb la perfSONAR UI, cal accedir a la pestanya Playground.

Per a provar el nostre servei, haurem de posar la URL del servei a la barra que per defecte ja inclou una adreça, a l'exemple es realitza una consulta al servei de lookup del perfHOST, per tant s'utilitza la URL d'entrada al servei que és:

<http://147.83.118.250:8180/geant2-java-xml-ls/services/LookupService>

Un cop seleccionat el host, cal seleccionar al desplegable situat a l'esquerra l'opció 'echo query' de manera que l'apartat query quedarà escrit amb la consulta predefinida.

Un cop seleccionat és moment d'executar la consulta amb l'opció 'Execute Query' del menú de la part esquerra. S'espera que la resposta sigui un missatge XML que contingui la frase '*This is the echo response from the service*' al camp de dades corresponent.

A continuació es mostra una imatge un cop rebuda la resposta del servei.

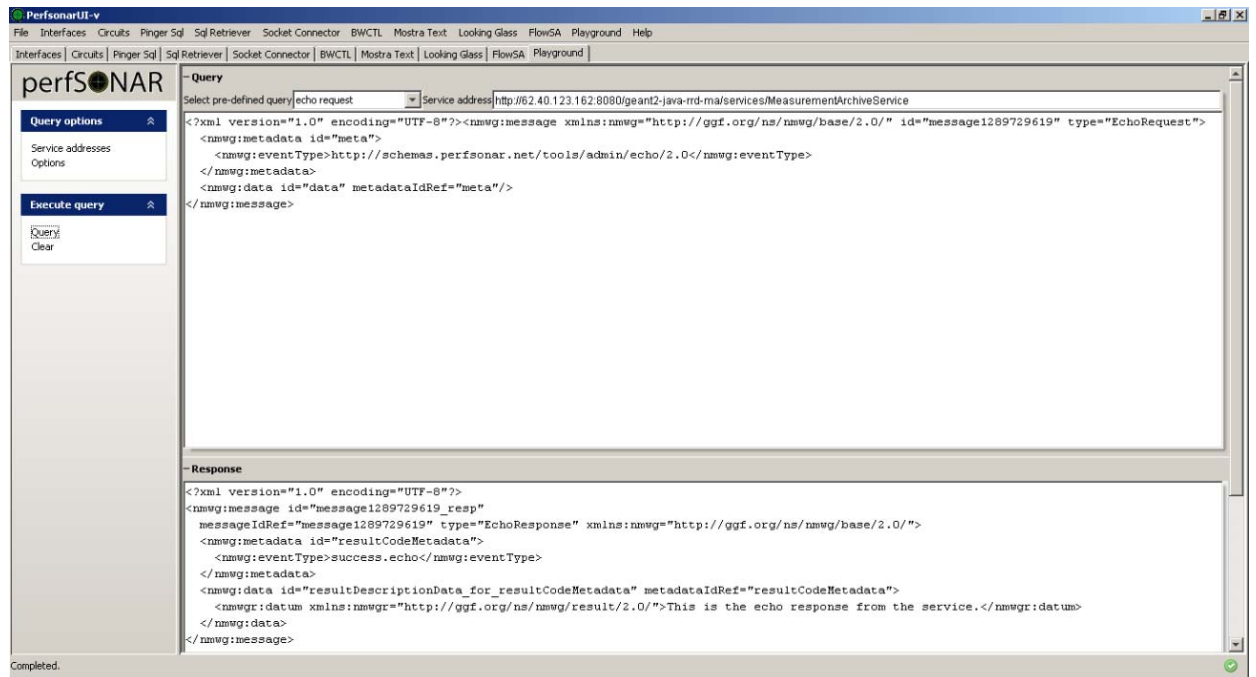


Fig D.2 Resultat de la prova de connexió al servei Lookup del perfhost

D.3 Configuració del servei

Per a configurar el *Lookup Service*, hem d'accedir a la plana web de configuració a través de la URL:

- <http://perfhost:8180/geant2-java-xml-ls/>

S'ha d'accedir a la opció '*Basic Configuration*' de l'apartat '*Service*' i introduir l'usuari i password d'entrada al servei, que per defecte són '*perfsonaruser*' i '*perfsonarpass*' respectivament.

A continuació comentarem els diferents apartats que s'han de col·leccionar per a una configuració bàsica tal i com apareixen a la versió 1.3 de la plana de configuració del servei:

- Authentication Configuration

En aquest apartat podem definir si volem activar l'autenticació per a determinats tipus de transaccions. Es defineix doncs quin ha de ser el servidor d'autenticació i el tipus de sol·licituds que requereixen autenticació. Aquesta opció farà que tan sols es processin determinats tipus de sol·licituds a sistemes de perfSONAR correctament identificats. En el nostre cas es decideix de deixar-la desactivada.

- LS Configuration

Aquesta opció consulta si volem registrar els nostres serveis amb un servei de *lookup* global, de manera que donaria una visió global del nostre servidor a la resta de serveis de *lookup* perfSONAR. En el nostre cas i donat que el nostre és un sistema de proves, no es selecciona aquesta opció per a no publicar els serveis que s'instal·lin a l'entorn global.

- eXist Configuration

En aquest punt es defineix la base de dades eXist que ha d'utilitzar el servei.

- *Enter the service name*
 - Es defineix el nom del servei de *Lookup* local.
- *Would you like to use gLS registration*

- Seleccionem un altre cop la opció no, ja que no volem registrar el nostre servidor de forma global.
- *Enter the service administrator's e-mail address:*
 - Indiquem al sistema una adreça de correu electrònic per a una possible comunicació respecte el sistema.
- *Specify the service user password for eXist*
 - Indiquem el password del usuari que es crearà per al servei de *lookup*. Es recomana utilitzar les opcions per defecte.
- *Would you like to initialize database structure?*
 - El sistema demana si volem inicialitzar l'estructura de la base de dades eXist, per tant la primera vegada que es configura el servei marcarem l'opció 'yes' i seleccionarem la opció 'no' si es modifica algun dels valors anteriors.
- *Set the registration interval (in seconds)*
 - Aquest valor indica al sistema l'interval de temps entre sol·licituds de registre al gLS. Per defecte es de dotze hores. No procedeix la modificació del valor ja que el sistema no es registrarà amb gLS.
- *Enter the service access point of hLS*
 - Indicar la ruta de l'accés al servei: En el nostre cas, <http://147.83.118.250:8180/geant2-java-xml-ls/services/LookupService>

D.4 Configuració del servei de *lookup* local per la resta de serveis

Els arxius de mesures i els punts de mesura de l'entorn perfSONAR permeten la configuració del registre envers un Lookup Service que permet doncs l'anunci dels serveis disponibles a la resta del sistema global de perfSONAR.

Acostuma a aparèixer la opció 'Do you wish to register with an LS?' a les planes de configuració dels seus serveis, i un cop es selecciona 'yes' apareixen les següents opcions:

- *Enter the service name*
Requereix que s'indiqui el nom del servei que s'ha instal·lat, aquest nom serà com estarà identificat el servei cap a la resta del sistema perfSONAR.

- *Enter a description for the service*
Es pot afegir una petita descripció del servei que s'ha instal·lat
- *Enter the service administrator's email address*
Es pot especificar l'adreça de contacte de l'administrador del servei per si es produeix qualsevol tipus de problema.
- *Enter the name of the organisation running this service*
El sistema dona també informació de la organització que proporciona aquest servei.
- *Enter the LS URL*
Aquí definim amb quin servei LS es vol registrar el servei, cal no confondre-la amb la URL de l'apartat de configuració, aquesta entrada apuntarà cap a 'localhost' sempre que el servei sigui instal·lat al mateix equip que el servei de lookup. En el nostre cas <http://localhost:8180/geant2-java-xml-ls/services/LookupService/>
- *Set the registration interval (milliseconds)*
Aquest valor estableix l'interval de temps entre missatges de registre, serveix per assegurar que si el servei està arrencat, el lookup service tindrà constància de la disponibilitat del mateix. El valor per defecte de 43200000 mil·lisegons (12 hores) és adequat.

ANNEX E. LLISTA DE SERVIDORS GLOBALS DE PERFSONAR

Donades les característiques dels serveis de lookup i la intenció de que el servei funcioni en un entorn multi-domini, els serveis de lookup han de replicar la informació dels serveis disponibles amb algun dels servidors globals. A continuació s'afegeix una llista dels serveis de lookup globals (gLS) mantinguts per les entitats promotores de perfSONAR (internet2, esnet, rnp).

Donat que tant els serveis locals com els globals han de compartir la informació entre ells, les llistes de servidors es publiquen a la següent URL:

<http://www.perfsonar.net/gls.root.hints>

Durant la realització d'aquesta memòria el contingut d'aquesta llista era el que es mostra a continuació:

```
http://ndbl.internet2.edu:9991/perfSONAR_PS/services/gLS
http://ndbl.internet2.edu:9992/perfSONAR_PS/services/gLS
http://ps1.es.net:9990/perfSONAR_PS/services/gLS
http://ps4.es.net:9990/perfSONAR_PS/services/gLS
http://ls.monipe.rnp.br:9990/perfSONAR_PS/services/gLS
http://xenmon.fnal.gov:9995/perfSONAR_PS/services/gLS
http://ls.perfsonar.pionier.net.pl:9990/perfSONAR_PS/services/gLS
http://ps1.jp.apan.net:9990/perfSONAR_PS/services/gLS
http://moonshine.damsl.cis.udel.edu:9991/perfSONAR_PS/services/gLS
```

Aquests serveis estan geogràficament ubicats entre Estats Units, Brasil, Polònia i Japó.

ANNEX F. INSTAL·LACIÓ I CONFIGURACIÓ DEL *AUTHENTICATION SERVICE*

F.1 Instal·lació del servei

En aquest apartat es comenta la instal·lació del servei d'autenticació que proporciona el sistema perfSONAR.

Per a instal·lar-lo, s'haurà d'executar amb privilegis d'administrador la següent sentència:

```
# apt-get install perfsonar-java-as
```

Es descarreguen a continuació tots els paquets i arxius necessaris del servei i a continuació comença la instal·lació. La instal·lació finalitza amb un reinici automàtic del servei Apache Tomcat.

F.2 Verificació de la instal·lació

Per verificar la instal·lació del servei s'ha d'accedir via web a la següent adreça:

- <http://147.83.118.250:8180/geant2-java-as/testdepl.jsp>

Fent clic al botó 'Test', el servei provarà de connectar amb l'entrada del servei i ens informarà de si s'ha pogut connectar adequadament.

F.3 Problemes trobats durant la instal·lació del servei

Un cop instal·lat el servei d'autenticació, al intentar executar la plana inicial '*index.jsp*' ens apareix el següent codi d'error:

```
org.apache.jasper.JasperException: Unparseable date: "Sat Jun 12  
19:45:58 GMT+01:00 2010"
```

```
org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:460)
```

```
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:355)
```

```
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:32
```

9)

```
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:265)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
```

Aquest error ens indica que el codi no pot donar format a la data correctament per mostrar-la través d'un navegador web. S'opta per examinar el codi i es busca la part on es defineix la modificació de la data. Al final cal realitzar una petita modificació al codi per a que tot i que la data no es mostri en el format desitjat, almenys es pugui utilitzar la interfície web.

Sent el codi de la plana 'index.jsp' resultant el que es mostra a continuació:

```
<%@ page
import="perfSONARWebAdmin.auxpestanyary.PerfsonarProperties,perfSONARWebAdmin.auxpestanyary.PerfsonarPropertiesImpl,perfSONARWebAdmin.auxpestanyary.PerfsonarProperty,java.io.File,java.io.FileInputStream,java.io.FileOutputStream,java.io.IOException" %>

<html>
<head>
<title>perfSONAR Service Web Administration page</title>
<link rel="stylesheet" type="text/css" href="main.css"/>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script src="Service_Admin.js"></script>

<style type="text/css">
<!--
a {
}
.style3 {
color: #8585c1;
font-size: 13px;
}
.style3 a:link {
color: #8585C1;
cursor: pointer;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
}
.style3 a:active {
color: #117721;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
}

.style3 a:visited {
color: #8585C1;
cursor: auto;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
/*font-weight: bold;*/
}
.style3 a:hover {
```

```

color: #8585C1;
cursor: pointer;
text-decoration: underline;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
}
-->
</style>
</head>
<body>
<%
    java.text.DateFormat dateFormat = new
    java.text.SimpleDateFormat("EEE MMM dd HH:mm:ss zzz yyyy");
    java.util.Date now= new java.util.Date();
    //La plana espera agafar la data desde la variable date, //però la
    //operació now.toString() falla, de manera que la //comentem i
    //assignem a la data directament l'objecte now
    // java.util.Date date= dateFormat.parse(now.toString());
    java.util.Date date = now;
    String confPath = "WEB-INF/classes/perfsonar/conf/";
    String ServicePropertiesPath = confPath+ "service.properties";
    ServletContext sc = getServletContext();
    String ServicePath = sc.getRealPath("/");
    PerfsonarProperty property = new PerfsonarProperty();
    PerfsonarProperties serviceProperties = new
    PerfsonarPropertiesImpl();
    serviceProperties.loadProperties(new FileInputStream(ServicePath +
    ServicePropertiesPath));

    String serviceName=
    ServiceProperties.getProperty("service.r.service_name").getValue()
    .trim();
    String serviceVersion=
    serviceProperties.getProperty("service.r.service_version").getValu
    e().trim();

%>
<div class="adminPanel">
<div class="banner">
    <table border="0" align="left" cellpadding="0" cellspacing="0">
<tr>
<td>
        <a href="welcome.htm" target="main"></a>
</td>
<td>
<table>
<tr> <td id="date"><%=date.toString()%></td></tr>
<tr> <td id="serviceName"><%=serviceName %>
&nbsp;  <%=serviceVersion%></td></tr>
</table>
</td>
</tr>
</table>
</td>

```



```

</tr>
</table>
</div>
<div class="mainWindow">
<div class="sidebar">
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<th>
<a href="welcome.htm" class="button" target="main">Admin home &nbsp;  
&nbsp;   &nbsp;  </a>
</th>
</tr>
<tr>
<td>
<p>
<span class="navtexheading">Basic
<br>
Configuration
</span>
</p>
<p>
<span class="navtex"><a href="Wizard" target="main">Service</a></span>
<br>
<a href="testdepl.jsp" target="main">Test</a>
</p>
</td>
</tr>

<tr>
<td>
<p>
<span class="navtexheading style3">Advanced
<br>
Configuration
</span>
</p>
<p>
<span class="navtex style3"><a href="ServiceAdmin?serviceProperties"
target="main">Service</a>
<br>
<a href="ServiceAdmin?loggingProperties" target="main">Logging</a>
<br>

</span>
</p>
</td>
</tr>
</table>

</div>
<div class="centerWindow">
<object data="welcome.htm" type="text/html" name="main" width="99%"
height="100%">
</object>

```

```
</div>
</div>
<div class="footer">
PERFORMANCE Service-Oriented Network monitoring ARchitecture &nbsp;
</div>
</div>
</body>
</html>
```

ANNEX G. INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI RRD-MA

G.1 Problema amb la instal·lació de la llibreria librrd2

Cap al mes d'Agost de 2010, al trobar problemes de funcionament del servei eXist db, es decideix reinstal·lar el servei Tomcat junt amb la base de dades per a tenir una configuració adequada.

Trobem doncs al instal·lar el paquet geant2-java-rrd-ma depèn del servei librrd2 que no és disponible per a la instal·lació als repositoris de Debian, ja que ha aparegut la versió 4 de la llibreria.

Es consulta a la llista de correu si hi ha algun mètode alternatiu per a la instal·lació d'aquest servei i rebem les següents instruccions:

You can install:

https://wiki.man.poznan.pl/images/perfsonar-mdm/files/perfsonar-java-rrd-ma/3.3/perfsonar-java-rrd-ma_3.3rc4_all.deb

But before you do that, you must do the following instructions (in case you have not done it before):

```
1)Add perfsonar-mdm-3.1 repository-execute the following commands:
# cd /etc/apt/sources.list.d
# wget http://downloads.perfsonar.eu/repositories/deb/perfsonar-mdm-3.1.list
# wget http://downloads.perfsonar.eu/repositories/perfsonar.asc
# apt-key add perfsonar.asc
# apt-get clean
# apt-get update
```

but in the list file change 'old-stable' to 'unstable' in both lines. There is no documentation for users to the new rc and some things changed so be careful of using it. But if you have already installed exist db with some data there will be no problem to use this new service. Please notice that url has changed from geant2-java-rrd-ma to perfsonar-java-rrd-ma.

Al seguir les indicacions rebudes des de la llista de correu, trobem que el paquet release candidate també té dependència de la llibreria librrd2 per tant, s'ha de buscar un altre alternativa.

Per a realitzar una instal·lació el més semblant a la que s'indica a la Guia d'administració, es decideix crear un repositori local per a publicar la llibreria

librrd2, que es troba a Internet, per a que el servei aptitude intenti descarregar el paquet des de el nostre repositori.

S'utilitza l'eina *reprepro*, que permet la creació de repositoris a partir d'un servidor web.

Accedim a la carpeta d'un recurs web que tinguem disponible i hi creem les següents carpetes:

```
# mkdir -p apt
# mkdir -p /apt/conf
# mkdir -p /apt/incoming
```

Un cop creats els directoris cal crear l'arxiu de configuració que indicarà el tipus de paquets que contindrà el repositori. Cal crear un arxiu de configuració anomenat '*distributions*', per a aquest repositori s'utilitza el següent arxiu:

```
Origin: perfrepo
Label: perfhost repo
Suite: stable
Codename: lenny
Version: 3.1
Architectures: amd64
Components: main non-free contrib
Description: perfhost repo for rrdtool
```

Per a publicar el paquet librrd2, cal executar la següent sentència

```
# reprepro -Vb . include lenny librrd2
```

Posteriorment s'ha d'afegir a l'arxiu */etc/apt/sources.list* una línia en la que definim el repositori local com la que es mostra a continuació:

```
deb http://localhost/repositori/apt/ lenny main
```

D'aquesta manera un cop s'intenta instal·lar qualsevol dels serveis, l'oficial i la versió '*release candidate*', es podria instal·lar el servei correctament.

El mètode seguit es va comentar amb el sr. Piotr Pikusa per a que es valori la inclusió el paquet librrd2 al repositori de perfSONAR per a facilitar aquesta tasca als usuaris.

G.2 Instal·lació del servei

Per a la instal·lació del servei s'ha d'executar la següent sentència en una consola de comandes com a usuari 'root'

```
# apt-get install perfsonar-java-rrd-ma
```

Un cop es reiniciï el servei Tomcat, el nou servei ja estarà instal·lat a l'equip.

G.3 Comprovació del servei

Per a configurar el servei, hem d'accedir a la plana web de configuració a través de la url:

<http://perfhost:8180/geant2-java-rrd-ma/>

Com ja s'ha comentat anteriorment, cal accedir a la opció 'Service' del menú 'Basic Configuration'. A continuació es detallen les opcions de configuració tal i com apareixen a la plana de configuració corresponent a la versió 3.1 del servei.

G.4 Configuració del servei

Per a configurar el servei, hem d'accedir a la plana web de configuració a través de la url:

- <http://perfhost:8180/geant2-java-rrd-ma/>

Com ja s'ha comentat durant la configuració d'altres serveis, cal accedir a la opció 'Service' del menú 'Basic Configuration'. El sistema demana nom d'usuari i contrasenya, que per defecte són 'perfsonaruser' i 'perfsonarpass' respectivament

A continuació es detallen les opcions de configuració tal i com apareixen a la plana de configuració corresponent a la versió 3.1 del servei.

- eXist Configuration
 - *Enter the URI location of the eXist database:*

El servei demana l'adreça URI d'entrada a la base de dades eXist. El valor per defecte en el nostre cas és l'adreça correcta: xmldb:exist://localhost:8180/exist/xmlrpc

- *Enter the service username for the eXist user*

En aquest punt definim el nom d'usuari que es generarà a la base de dades eXist. Per a assegurar un correcte funcionament de l'eina es recomana utilitzar el nom que hi apareix per defecte → *rrdmaservice*

- *Enter the password for the eXist user*

Es defineix aquí el *password* de l'usuari del servei. En aquest cas es recomana també utilitzar el valor per defecte que s'indica a la guia d'instal·lació de perfSONAR → *rrdmaconfig*

- *Do you wish to create a user with this name?*

Aquesta opció determina la creació o no de l'usuari del servei a la base de dades eXist. Es recomana utilitzar la opció 'yes' la primera vegada que s'instal·la el servei, de manera que es crearà l'usuari i el seu perfil amb el password que es defineixi al formulari que es mostra si seleccionem aquesta opció.

- *Enter the full path of the configuration file:*

Aquest valor defineix la ruta al sistema d'arxius del disc de l'arxiu XML de configuració del servei, en el qual s'hi defineixen les interfícies de les que es disposen dades

- *Do you wish to change the eXist Admin password?*

La configuració ens permet modificar el password de l'usuari administrador del servei eXist, però en cas de que es vulgui modificar, es recomana la modificació a través de la plana de configuració del servei eXist descrita annex C.

- *Do you wish to enable Authentication?*

Si es selecciona la opció 'yes', s'ha d'indicar el servei *Authentication Service* que s'encarregarà de l'autenticació així com el tipus de sol·licituds que requeriran autenticació per a ser servides.

- *Do you wish to register with a LS?*

Aquest apartat permet la publicació de la disponibilitat del servei a un servidor de Lookup que es defineixi. Al seleccionar la opció 'yes' apareixen diversos camps de text. A l'annex G es detallen els valors que s'hi poden adjuntar en aquests camps.

- *Enter the password for logging in to the Admin pages:*

El servei torna a demanar el password d'accés a l'apartat de configuració per seguretat. Aquesta contrasenya és per defecte *'perfsonarpass'*.

G.5 Generació de l'arxiu de configuració del servei

Aquest arxiu és el que defineix les interfícies de les que el servei emmagatzema dades, i per tant aquelles dades a les que hi podem accedir. Es realitza mitjançant un arxiu XML que relaciona les interfícies que tenim definides amb la ubicació de les bases de dades que contenen la informació. En aquest apartat es mostrarà la manera de definir una interfície de la que s'està monitoritzant la utilització tant d'entrada com de sortida.

La definició es divideix en dos blocs:

- Bloc de Meta-dades: Es defineix informació relativa a les dades que es mostraran, com el nom de l'equip, l'adreça IP de la interfície, el nom d'aquesta, una breu descripció o la capacitat de l'enllaç.
- Bloc de dades: És en aquest bloc en el qual s'indica la ubicació de les dades i la informació rellevant per a que el servei pugui realitzar la consulta a la base de dades. S'hi defineixen majoritàriament paràmetres.

Mostrem a continuació com es defineix a l'arxiu:

```
<!--Inici del Bloc de Meta-dades-->
<nmwg:metadata id="meta1"> <!-- Nom del bloc de meta-dades -->
  <netutil:subject id="subj1">
    <nmwgt:interface>
      <nmwgt:hostname>Ubuntu</nmwgt:hostname> <!--Nom del host-->
      <nmwgt:ifAddress type="ipv4">147.83.118.251</nmwgt:ifAddress><!--@IP -->
      <nmwgt:ifName>Eth0</nmwgt:ifName> <!--Nom de la interfície -->
      <nmwgt:ifDescription>Utilitzacio @ Ubuntuhost</nmwgt:ifDescription>
      <nmwgt:direction>in</nmwgt:direction>
      <nmwgt:capacity>1000BaseT</nmwgt:capacity> <!-- Tipus d'enllaç -->
    </nmwgt:interface>
  </netutil:subject>
  <nmwg:eventType>http://ggf.org/ns/nmwg/characteristic/utilization/2.0</nmwg:ev
ventType></nmwg:metadata>
<!--Fi del bloc de Metadades-->

<!--Inici del Bloc de dades-->
<nmwg:data id="data1" metadataIdRef="meta1"> <!--Bloc de Dades 'data1' fa
referencia al bloc de meta-dades 'meta1'>
<nmwg:key>
  <nmwg:parameters>
    <nmwg:parameter name="file">/usr/lib/perfsonar/services/geant2-java-rrd-
ma/WEB-INF/samples/rrd/basededades.rrd</nmwg:parameter><!--Ruta de l'arxiu
RRD-->
    <nmwg:parameter name="dataSource">inoctets</nmwg:parameter>
```

```

<nmwg:parameter name="valueUnits">Bps</nmwg:parameter>
<nmwg:parameter
name="eventType">http://ggf.org/ns/nmwg/characteristic/utilization/2.0</n
mwg:parameter>
- <!--A partir d'aquest punt es poden afegir aquests valors opcionals que
donen informació de la base de dades RRD -->
<nmwg:parameter name="dataSourceStep">300</nmwg:parameter>
<nmwg:parameter name="dataSourceType">ABSOLUTE</nmwg:parameter>
<nmwg:parameter name="dataSourceHeartbeat">400</nmwg:parameter>
<nmwg:parameter name="dataSourceMinValue">0</nmwg:parameter>
<nmwg:parameter name="dataSourceMaxValue">10000000</nmwg:parameter>
</nmwg:parameters>
</nmwg:key>
</nmwg:data>
<!--Fi del bloc de dades-->

```

G.6 Creació de les bases de dades RRD

A l'hora de crear una base de dades RRD s'han de tenir en compte diversos factors, donada la naturalesa d'aquest tipus de bases de dades. Ja s'ha comentat anteriorment que aquest tipus de base de dades emmagatzema el valor real durant el temps que es defineixi per a després reutilitzar els valors antics utilitzant una funció de consolidació.

Per a la creació d'una base de dades RRD s'han de definir els següents paràmetres:

- Nom de la base de dades: Cal escollir un nom per a l'arxiu que tingui extensió .rrd.
- Fonts de dades: S'utilitza una sentència per a cadascuna de les fonts de dades a crear, en les que s'hi indica:
 - Nom de la font
 - Tipus de valor: COUNTER, GAUGE, DERIVE, ABSOLUTE
 - Pas entre mesures: El valor en segons entre mesura i mesura
 - Valor mínim
 - Valor màxim
- Arxiu de Round Robin: Es correspon als arxius de mesures antigues que es van creant a mida que passa el temps.
 - Funció de Consolidació: MIN, MAX o AVERAGE
 - Xff: Factor d'expedients X, s'utilitza per a discernir el valor final a emmagatzemar en cas de que alguna de les mides tingui com a resultat un valor desconegut.
 - Nombre de mostres consolidades.
 - Nombre de mostres desades.

Per al servei d'exemple instal·lat, es crea una base de dades amb la següent sentència:


```
1. rrdtool create db.rrd
2. --step 300
3. DS:ds0:COUNTER:300:0:4294967296
4. DS:ds1:COUNTER:300:0:4294967296
5. RRA:MAX:0.5:1:600
6. RRA:MAX:0.5:6:600
7. RRA:MAX:0.5:24:600
8. RRA:AVERAGE:0.5:288:600
9. RRA:AVERAGE:0.5:1:600
10. RRA:AVERAGE:0.5:24:600
11. RRA:MAX:0.5:288:600
12. RRA:AVERAGE:0.5:6:600
```

En aquesta base de dades s'emmagatzemarà el valor del comptador de bytes transmesos i rebuts que s'obtindrà a partir de consultes SNMP.

La línia 1 defineix la creació d'una base de dades anomenada db.rrd al directori des de on estiguem cridant l'aplicació. La línia 2 defineix l'interval entre el qual es reben dades, en aquest cas es rep una mesura cada 300 segons, 5 minuts. Les línies 3 i 4 defineixen el que es vol emmagatzemar a la base de dades, les fonts de dades o *datasources*, es defineixen utilitzant el següent esquema:

DS:nom_variable:TIPUS_DADES:batec:min:max

A Destacar el camp TIPUS_DADES, que defineix el tipus de dades que pot emmagatzemar l'eina. Els tipus de dades son:

- **COUNTER:** El camp que hi desem es com un comptador, el seu valor s'incrementa a cada període.
- **DERIVE:** És també un comptador però admet nombres negatius i es idoni per a dades que no sempre augmentin el seu valor, per exemple per si volguéssim veure com varia l'espai en disc disponible d'un servidor, que pot augmentar o disminuir.
- **ABSOLUTE:** Aquest tipus, també permet desar la variació del valor, però assumeix que el valor previ està marcat a 0. La diferència entre el valor actual i el previ sempre és igual al valor actual. Només desa el valor actual dividit entre el interval entre mesures.
- **GAUGE:** Aquest tipus de dades no salva la taxa de variacions. Salva el valor actual per si mateix. No hi ha divisions ni càlculs. El consum de memòria d'un servidor n'és un típic exemple d'aquest tipus de dades.

El batec defineix el temps després del qual si no s'ha rebut un valor per a afegir a la base de dades, es considerarà desconegut. És a dir, si en 300 segons no es

rep una mesura, el sistema encara l'espera durant 300 segons abans d'assignar a la mesura un valor desconegut.

Les línies 5 a 12 defineixen el tipus d'arxius temporals que es generen, s'utilitza la següent estructura per a definir-los.

```
RRA:FUNCIO_CON:XFF:step:FILES
```

RRA és la paraula clau per a la definició dels arxius RRA, la segona opció és la funció de consolidació, que pot ser AVERAGE (Mitjana) MINIMUM, MAXIMUM i LAST (darrera). Aquí és on intervé el concepte de CDP (Punt de dades consolidades).

Un punt CDP és el resultat de passar per la funció de consolidació el nombre de mesures indicat per 'step'. És a dir que un cop es recullen 'step' mesures es passen per la funció de consolidació i d'aquests conjunts se'n desen tants com indiqui el valor FILES.

En el nostre exemple, a la línia 5, s'agafen blocs d'una mesura, se'n fa la mitja i es desen 600 d'aquests blocs. En aquest cas, s'emmagatzemen les 600 darreres mesures. La mesura 601 sobreesciu el primer valor.

A la línia 10, per exemple, es defineix un arxiu que es fa la mitja de 24 mesures (1 mesura cada 5 minuts, implica la mitja de 2 hores de mesures) i es desen 600 d'aquests blocs, per tant s'està cobrint al voltant de 50 dies. Un cop han passat els 50 dies, la mitja de les primeres dues hores següents sobreesciu el valor del principi.

A la línia 11, es defineix un arxiu que agafa el màxim entre les 288 mesures de tot un dia, i n'emmagatzema un total de 600, de manera que es té el valor màxim durant 600 dies. El dia 601 sobreescrirà el valor del primer dia.

A l'hora de definir les fonts de dades (ds), es determina que el tipus de variable és un comptador, que el valor mínim d'aquest serà zero i el màxim es el valor màxim que tindrà un comptador de trenta-dos bits. Indicar que les funcions de consolidació que es defineixen son MAX i AVERAGE donades les característiques de les mesures a emmagatzemar. Interessa tenir constància si en algun moment la utilització ha estat 0 i per tant s'ha produït una caiguda de l'enllaç o quan s'apropa al màxim de la interfície, on es podria provocar una saturació de l'enllaç.

G.7 Verificació de les mesures

Per a verificar un correcte funcionament del sistema, s'ha d'utilitzar la interfície perfSONAR UI que per defecte accedeix a la pestanya 'Interfaces' on mostra les dades disponibles a serveis d'arxiu de mesures.

Un cop seleccionat el servidor que es vol consultar, s'ha de seleccionar la data i el tipus de mesures que es vol obtenir, el servei contacta amb el servidor i ens mostra aquelles mesures seleccionades, tal com es pot veure a la imatge 3.4 més endavant.

Les dades que conté el servidor que s'ha instal·lat durant la realització d'aquest treball, s'obtenen a partir de diferents transmissions de dades entre dos equips.

Per a verificar que les dades que es mostren a la gràfica son valors reals, s'ha utilitzat l'eina monitora 'IPTraf'[18] que mostra la velocitat de transferència instantània, i comparant-los amb els valors mitjans que s'observen a la interfície gràfica de perfSONAR s'extrau que són indicats (veure imatge 3.5).

Per a generar tràfic, s'aprofita la opció `-s` del ping del sistema per a enviar pings amb la major quantitat de dades que ens permet el sistema operatiu, utilitzant la següent comanda:

- `ping 147.83.118.251 -s 65507`

Un cop els equips s'estan intercanviant missatges ICMP, observem que la consola de perfSONAR i l'aplicació IPTraf coincideixen en que la taxa d'entrada és al voltant de 17 Mbps, tal i com es pot observar en les imatges a continuació.

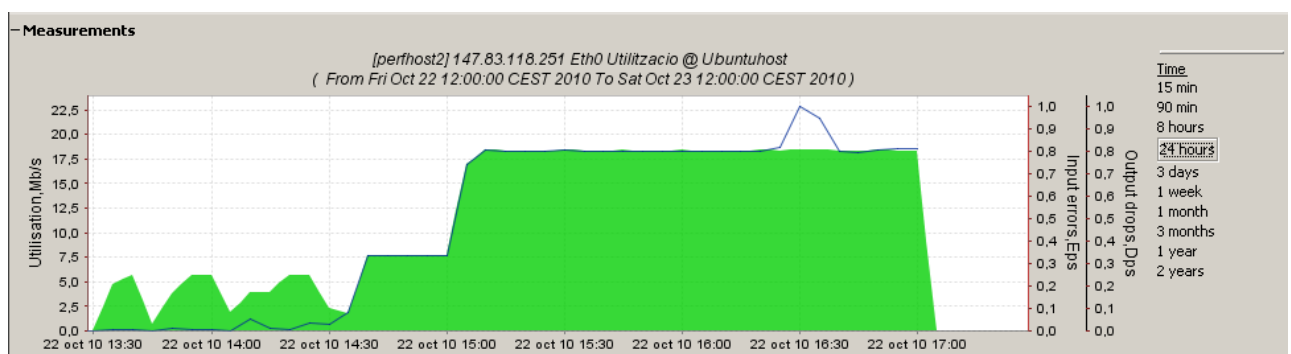


Fig. G.1 Gràfic extret de la consola perfSONAR UI, es mostra la utilització de la interfície `eth0` del servidor de proves

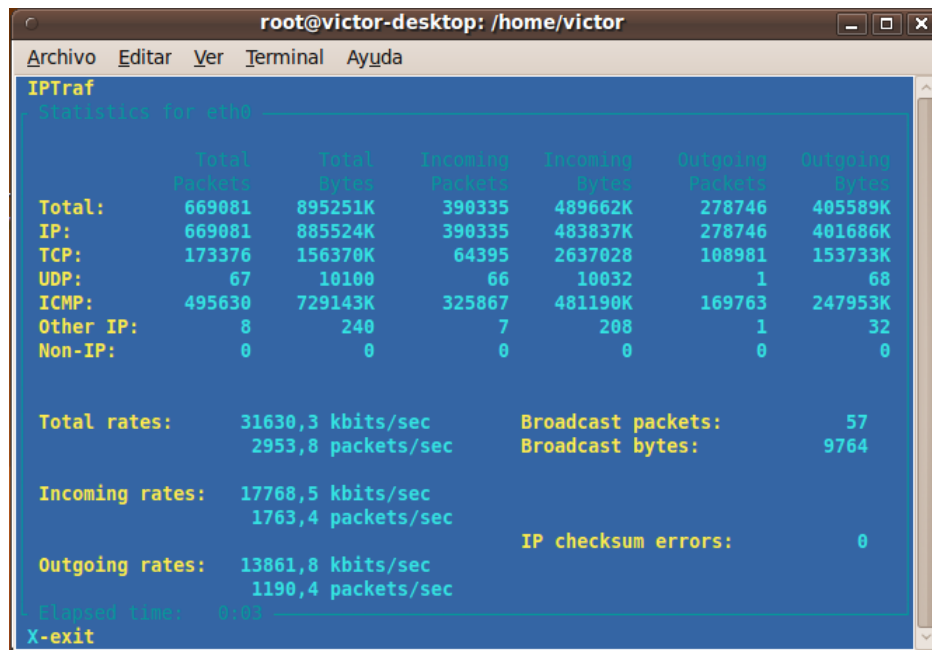


Fig. G 2 Pantalla corresponent al software IPtraf que ens mostra estadístiques dels dispositius de xarxa del sistema, en aquest cas la interfície eth0

S'ha observat que els valors de la taxa de sortida mostrats a IPtraf no concorden amb els que s'indiquen a la perfSONAR UI. Això es deu a un problema amb la configuració de les targetes de xarxa virtuals que utilitza el servidor que rep el ping.

Aquesta investigació quedaria lluny de l'abast d'aquest projecte, però s'aprecia un comportament estrany de l'eina IPtraf al accedir a la màquina receptora (màquina virtual) a través de VNC, el comptador d'ambdues màquines augmenta la seva taxa, i per tant es creu que hi ha algun tipus de comportament irregular amb els comptadors de tràfic al treballar amb equips virtuals.

Per a la verificació de les dades que s'han anat desant a bases de dades RRD i SQL, s'ha generat tràfic controlat i constant a partir de l'enviament de trames *ICMP Echo Request* habilitant la opció '-s 65507' que enviava trames amb un total de 65515 bytes de dades amb un interval, predefinit per l'eina *ping*, de 1 segon entre cada paquet *request*. Les trames s'envien des de la màquina física cap a l'adreça de la màquina virtual.

S'obté doncs que per cada terminal executant un ping d'aquestes característiques, tenim:

$$65515 \frac{\text{bytes}}{\text{paquet}} \times 1 \frac{\text{paquet}}{s} \times 8 \frac{\text{bits}}{\text{byte}} = 524120 \text{bps} \approx 0.5 \text{Mbps} \quad (\text{I.1})$$

S'envien un total de 22 de finestres executant ping, de manera que s'obté una utilització de:

$$0.5 \text{Mbps} \times 22 \text{ter min als} = 11 \text{Mbps} \quad (\text{I.2})$$

I al ser una comunicació d'anada i tornada, ja que un *echo request* es seguit per un *echo reply*, tenim doncs que tenir en compte ambdós sentits fent un total de 22Mbps.

A les imatges a continuació, es pot apreciar la diferència dels marcadors respecte aquests valors aquí indicats, a la part superior esquerra es pot trobar l'estat de l'eina iptraf a la màquina virtual, sent doncs la part dreta l'estat de l'eina la corresponent a la tarja del servidor físic.

Un cop es realitza una connexió VNC a la màquina virtual s'aprecia un augment considerable del tràfic a la pantalla d'iptraf del servidor físic.

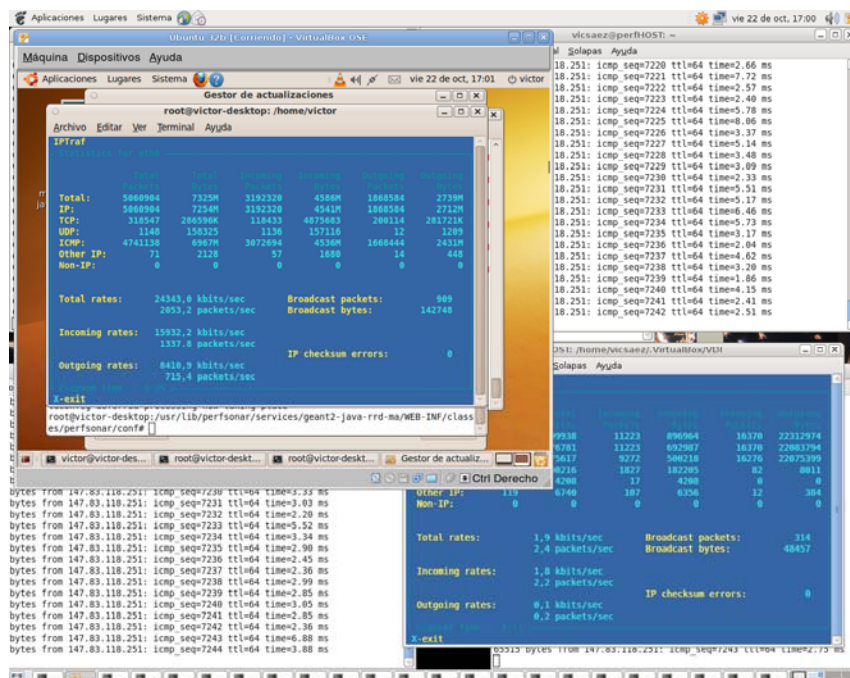


Fig G 3 Comparació del tràfic d'entrada i sortida a les interfícies virtual i física dels servidors sense connexió VNC.

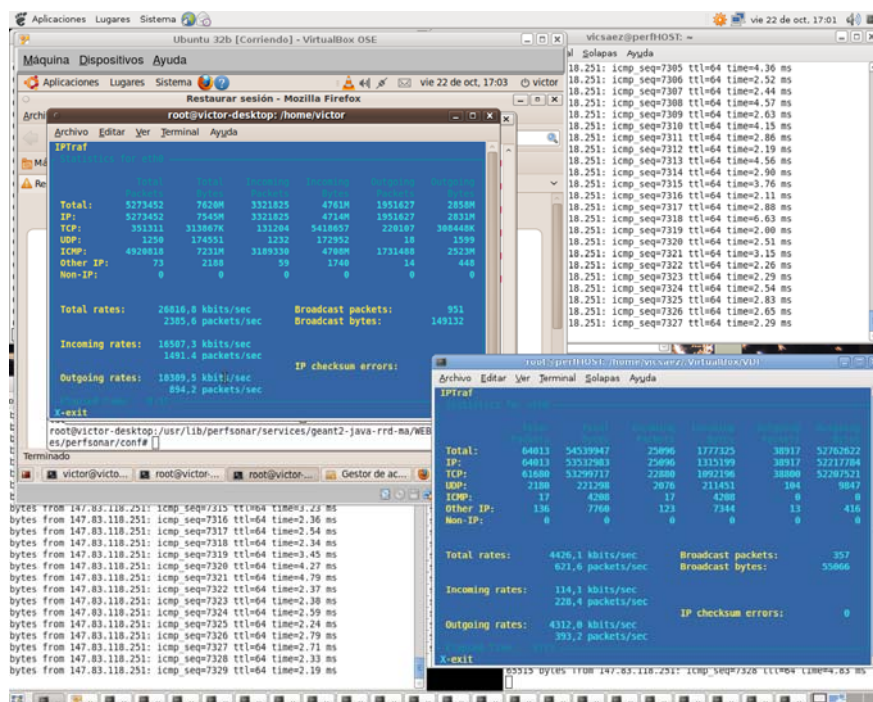


Fig G 4 Comparació del tràfic entrant i sortint de les targetes de xarxa virtual i física dels servidors amb connexió VNC.

Les taules a continuació mostren una comparativa dels valors que apareixen a les imatges:

Taula G.1. Tràfic observat a la màquina física

Maquina física	Sense VNC	Amb VNC
Tràfic entrada	0.1 kbps	114.1 kbps
Tràfic sortida	1.8 kbps	4312 kbps
Tràfic total	1.9 kbps	4426 kbps
Tràfic total teòric	≈22Mbps	≈22Mbps + Tràfic VNC

Taula G.2. Tràfic observat a la màquina virtual

Maquina virtual	Sense VNC	Amb VNC
Tràfic entrada	15.56 Mbps	16.12 Mbps
Tràfic sortida	8.21 Mbps	10.07 Mbps
Tràfic total	23.77 Mbps	26.19 Mbps
Tràfic total teòric	≈22Mbps	≈22Mbps + Tràfic VNC

S'aprecia als valors que en el cas de la màquina virtual, els valors son bastant més propers al que s'espera, almenys en quant a tràfic total, sent el tràfic d'entrada en el primer cas (sense VNC) superior al teòric.

Però no es el cas amb la màquina física, ja que segurament el tràfic destinat a la màquina virtual no s'envia a través de la interfície física, si no a través d'un adaptador de xarxa virtual. S'aprecia també a la interfície física del servidor la contribució de la connexió VNC, que genera bastant tràfic de xarxa.

Segons les dades desades al sistema perfSONAR, la utilització era propera als 17.5 Mbps, tot i que es comprova amb aquesta prova que el que mostra l'eina iptraf pot presentar certes diferències sobre els valors reals, i per tant també els comptadors SNMP que estem consultant podrien presentar valors reals amb alguna diferència, a més de possibles serveis que generin tràfic no controlat.

G.8 Script col·lector de dades

Aquest script s'ha utilitzat per a la obtenció i publicació de dades utilitzant les eines que s'han desenvolupat.

```
#!/bin/sh

a=0
while [ $a==0 ]; do

#Consultes SNMP
snmpwalk -Ov -v2c -c perfhost 147.83.118.251
interfaces.ifTable.ifEntry.ifInOctets.2 > /home/victor/TEST/snmp_exit
snmpwalk -Ov -v2c -c perfhost 147.83.118.251
interfaces.ifTable.ifEntry.ifOutOctets.2 > /home/victor/TEST/snmp_exit2
snmpwalk -Ov -v2c -c perfhost 147.83.118.251
interfaces.ifTable.ifEntry.ifInErrors.2 > /home/vicsaez/TEST/snmp_exit3
snmpwalk -Ov -v2c -c perfhost 147.83.118.251
interfaces.ifTable.ifEntry.ifInDiscards.2 >
/home/vicsaez/TEST/snmp_exit4

#Aïllar el valor del contador SNMP amb awk
inOctets=`awk '{print $2}' FS=":" /home/victor/TEST/snmp_exit`
outOctets=`awk '{print $2}' FS=":" /home/victor/TEST/snmp_exit2`
inErrors=`awk '{print $2}' FS=":" /home/vicsaez/TEST/snmp_exit3`
inDiscards=`awk '{print $2}' FS=":" /home/vicsaez/TEST/snmp_exit4`

echo "InOctets: $inOctets OutOctets= $outOctets"
N=`date +%s`
echo $N
```

```
echo "rrdtool update /usr/lib/perfsonar/services/geant2-java-rrd-
ma/WEB-INF/classes/perfsonar/conf/db.rrd $N:$inoctets:$outoctets"

#Update de la base de dades RRD amb rrdtool
rrdtool update /usr/lib/perfsonar/services/geant2-java-rrd-ma/WEB-
INF/classes/perfsonar/conf/db.rrd $N:$inoctets:$outoctets
java -jar /home/victor/Descargas/sqlr.jar $inoctets $outoctets

#Esperar 5 minuts entre mesura i mesura
echo "Esperant 300s"
perl -e 'sleep 300 - time % 300'
echo "Fet!"

#Update de base de dades SQL amb fetcher.jar (paquet jar executable).
java -jar /home/victor/Descargas/fetcher2.jar $inoctets >>
/home/victor/Descargas/fetcher.txt

done
```


ANNEX H. CODI PER A LA PUBLICACIO DE MESURES A TRAVES DEL SERVEI RRD-MA

A continuació es pot trobar el codi utilitzat per a enviar el resultat d'una mesura SNMP a través de un missatge *store* a l'entrada del servei *RRD-MA*.

```
package fetcher;
import java.net.*;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.io.*;

public class SOAPster {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String valor=args[0];
        Calendar c = GregorianCalendar.getInstance();
        long timestamp;
        timestamp=c.getTimeInMillis()/1000;

        try {
            String xmldata =
                "<?xml version=\"1.0\" encoding=\"UTF-8\"?><soapenv:Envelope\n"
                + "xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"\n"
                + "xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"\n"
                + "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-\n"
                + "instance\"><soapenv:Body><nmwg:message id=\"msg4\"\n"
                + "type=\"MeasurementArchiveStoreRequest\"\n"
                + "xmlns:netutil=\"http://ggf.org/ns/nmwg/characteristic/utiliz\n"
                + "ation/2.0/\" xmlns:nmwg=\"http://ggf.org/ns/nmwg/base/2.0/\"\n"
                + "xmlns:nmwgt=\"http://ggf.org/ns/nmwg/topology/2.0/\">\n"+
                "    <nmwg:metadata id=\"meta1\">\n"+
                "        <netutil:subject id=\"subj1\">\n"+
                "            <nmwgt:interface>\n"+
                "                <nmwgt:hostname>Ubuntu2</nmwgt:hostname>\n"+
                "                <nmwgt:ifAddress\n"
                + "                    type=\"ipv4\">147.83.118.252</nmwgt:ifAddress>\n"+
                "                <nmwgt:ifName>eth0</nmwgt:ifName>\n"+
                "                <nmwgt:ifDescription>Utilitzacio\n"
                + "In</nmwgt:ifDescription>\n"+
                "            <nmwgt:direction>in</nmwgt:direction>\n"+
                "            <nmwgt:authRealm>TestRealm</nmwgt:authRealm>\n"+
                "            <nmwgt:capacity>100BaseT</nmwgt:capacity>\n"+
                "        </nmwgt:interface>\n"+
                "    </netutil:subject>\n"+
                "    <nmwg:eventType>http://ggf.org/ns/nmwg/characteristic/ut ili\n"
                + "zation/2.0</nmwg:eventType>\n"+
                "  </nmwg:metadata>\n\n"+
                "    <nmwg:data id=\"data1\" metadataIdRef=\"meta1\">\n"+
                "<nmwg:datum timeType=\"unix\"          timeValue=\"\"+timestamp+\"\"\n"
                + "value=\"\"+valor+\"\" valueUnits=\"Bps\"/>\n" +
```

```

"    </nmwg:data>\n"+
    "</nmwg:message></soapenv:Body></soapenv:Envelope>\n";

    //Create socket

    String hostname = "147.83.118.251";
    int port = 8180;
    InetAddress addr = InetAddress.getByName(hostname);
    Socket sock = new Socket(addr, port);

    //Enviem capçalera, indiquem el servei al que volem
    enviar el missatge
    String path = "/geant2-java-rrd-
    ma/services/MeasurementArchiveService";
    OutputStreamWriter wr = new
    OutputStreamWriter(sock.getOutputStream());

    BufferedWriter esc = new BufferedWriter(new
    OutputStreamWriter(sock.getOutputStream()));

    wr.write("POST "+path+" HTTP/1.0\n");
    wr.write("Content-Type: text/xml; charset=\n\"utf-
    8\"\n");
    wr.write("Accept: application/soap+xml,
    application/dime, multipart/related, text/*\n");
    wr.write("User-Agent: Axis/1.4\n");
    wr.write("Host: 147.83.118.251:8180\n");
    wr.write("Cache-Control: no-cache\n");
    wr.write("Pragma: no-cache\n");
    wr.write("SOAPAction: \n\n\n");
    wr.write("Content-Length: " + xmldata.length() +
    "\n\n");

    //Enviem les dades
    wr.write(xmldata);
    wr.flush();

    // Response
    BufferedReader rd = new BufferedReader(new
    InputStreamReader(sock.getInputStream()));
    String line;

    while((line = rd.readLine()) != null){
        System.out.println(line);
    }
    sock.close();
} catch (Exception e) {
    e.printStackTrace();
}
System.out.println("-----
    *");
}
}

```

ANNEX I. INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI SQL-MA

H.1 Instal·lació

Un cop instal·lat el motor de la base de dades, és el moment de la instal·lació del servei de perfSONAR, que es realitza executant la següent comanda, també amb privilegis d'administració:

- `apt-get install geant2-java-sql-ma`

Un cop finalitzada la instal·lació, el servei Apache Tomcat és reiniciat i ja es pot accedir a la plana de configuració a través de la següent URL:

- <http://147.83.118.251:8180/geant2-java-sql-ma/>

Si seleccionem la opció *Test* de l'apartat *Basic Configuration*, podrem realitzar una prova de connexió amb l'entrada del servei com la que s'ha descrit a l'apartat 2.3.5

H.2 Configuració del servei

Els dos arxius de mesures que proporciona perfSONAR tenen certes similituds a l'hora de la configuració, es comenten en aquest apartat aquells paràmetres que no s'han comentat a l'apartat de configuració del servei RRD-MA a l'annex G.

La configuració es realitza accedint a la següent URL

- <http://147.83.118.251:8180/geant2-java-sql-ma>

Podrem accedir al formulari a partir de la opció *Basic Configuration* del menú *Service*. El sistema demanarà l'usuari i *password* per defecte, '*perfsonaruser*' i '*perfsonarpass*' respectivament.

En aquest servei, s'ha d'especificar els fitxers ibatis que permetran la connexió amb la base de dades SQL, s'han de seleccionar accedint a l'apartat *SQL MA Admin*, que es troba al menú *Advanced Configuration*. La versió instal·lada, *pS java sql ma 2.1*, mostra una pantalla d'error a l'hora d'accedir a aquest directori, es pot consultar les accions realitzades per a intentar solucionar l'error al punt 3.2.9.

ANNEX J. CODI PER A LA PUBLICACIÓ DE MESURES D'UTILITZACIÓ AL SERVEI SQL-MA

El codi que es presenta a continuació rep per paràmetre els valors capturats a través de SNMP, consulta a una base de dades SQL els valors emmagatzemats a l'anterior mesura i calcula la utilització amb l'expressió 3.3. Posteriorment desa les mesures actuals per a desar-les després. El programa romandrà aturat mentre no rebi més mesures.

```
package sqler;
import java.sql.*;
import java.util.Calendar;
import java.util.GregorianCalendar;

    public class esecueler {
        Connection conexion;
        ResultSet tabla;
    public static void main (String args[]){
        String inoctets, outoctets;
        long in=Long.parseLong(args[0]);
        long out=Long.parseLong(args[1]);
        long oldin=0, oldout=0;
        ResultSet resultat;
        String insq, updq, olds;
        long aux=2^32;
        olds="select * from `perfsonar_ma`.`victor`";
        esecueler sql = new esecueler();

        //System.out.println("Utilització: "+util+" bps");

        if(sql.Connect()){
            resultat=sql.Select(olds);
            try {
                while(resultat.next()){
                    oldin=resultat.getLong("inoctets");

                oldout=resultat.getLong("outoctets");
                }
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                System.out.println("Error al obtenir els
                anteriors");
                e.printStackTrace();
            }

            updq="UPDATE `perfsonar_ma`.`victor` SET
            `inoctets` = '"+in+"',`outoctets` = '"+out+"' WHERE `victor`.`inoctets`
            ='"+oldin+" AND `victor`.`outoctets` ='"+oldout+" LIMIT 1";
            sql.Insert(updq);

            //32 bits --> contador 0 a 4294967296
```

```

        if (oldin>in){
            in=(aux-oldin)+in;
            System.out.println("Contador in ha donat
            la volta!");
            System.out.println("OLDin      "+oldin+"\nin
            "+in+"\nOLDout "+oldout+"\nout "+out);

        }else if(oldout>out){
            out=(aux-oldout)+out;

            System.out.println("Contador out ha donat
            la volta!");
            System.out.println("OLDin      "+oldin+"\nin
            "+in+"\nOLDout "+oldout+"\nout "+out);
        }else{
            in=in-oldin;
            out=out-oldout;
        }
        //insq="insert      into      victor      values
        ("+"in+", "+"out+")";

        //sql.Insert(insq);
        int util = ((in + out)*8)/(300);

        System.out.println("Util que serà ara 0
        "+util+"bps");
        System.out.println("Util Mbps:"+util/(1024^2));
        Calendar c = GregorianCalendar.getInstance();
        long timestamp;
        timestamp=c.getTimeInMillis()/1000;
        insq="insert      into      perfsonar_utilization
        (metadataaid, value, valueUnits, timeValue, ts)
        values      "+"('metal-
        test', "+"util+", 'Bps', "+timestamp+", '"+c.get(Cale
        ndar.YEAR)+"-"+(c.get(Calendar.MONTH)+1)+"-
        "+c.get(Calendar.DAY_OF_MONTH)+"
        "+c.get(Calendar.HOUR_OF_DAY)+"-"+c.get(Calendar
        .MINUTE)+"-"+c.get(Calendar.SECOND)+"')";
        System.out.println(insq);
        if(!sql.Insert(insq)){
            System.out.println("Fallo al fer insert");
        }

    }else{
        System.out.println("Error al crear la conexio");
        System.exit(1);
    }

    if(!sql.Disconnect()){
        System.out.println("Fallo al desconectar");
    }

}

public boolean Connect(){
    //true si conecta, false si no
    try {
        Class.forName("com.mysql.jdbc.Driver");

```

```

                                conexion =
        DriverManager.getConnection("jdbc:mysql://147.83
        .118.251/perfsonar_ma?user=root&password=c4st3ll
        d3f3ls");
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        System.out.println("ClassNotFoundException");
        //e.printStackTrace();
        return false;
    } catch (SQLException e) {

        e.printStackTrace();
        return false;
    }
    return true;
}

public ResultSet Select(String Query){
    Statement instruccion;
    try {
        instruccion = conexion.createStatement();
        tabla = instruccion.executeQuery(Query);

    } catch (SQLException e) {
        System.out.println("Fallo al crear Statement o
        Enviar Query");
        e.printStackTrace();
    }
    return tabla;
}

public boolean Insert(String Query){
    Statement instruccion;
    try{
        instruccion = conexion.createStatement();
        instruccion.executeUpdate(Query);
    }catch (SQLException e){
        System.out.println("Fallo al fer
        Insert/update");
        e.printStackTrace();
        return false;
    }
    return true;
}

public boolean Disconnect(){
    try {
        conexion.close();
    } catch (SQLException e) {
        System.out.println("Fallo al tancar Conexio");
        return false;
    }
    return true;
}
}

```

ANNEX K. INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI BWCTL-MP

K.1 Prerequisits

K.1.1 NTP

La instal·lació es realitza a partir de la següent comanda:

```
# apt-get install ntp
```

Per a configurar els servidors amb els quals es sincronitza, s'ha d'afegir una sèrie de línies a l'arxiu

```
# /etc/ntp.conf
```

Els servidors es poden consultar a [24], que conté informació sobre servidors horaris disponibles classificats per regió. A més dels servidors de Debian que hi ha predefinitos al servei, s'afegeixen els següents:

- 3.es.pool.ntp.org
- 0.europe.pool.ntp.org
- 3.europe.pool.ntp.org

K.1.2 Iperf

Per a la seva instal·lació cal descarregar l'aplicació que es proporciona en un paquet comprimit. Per tant, des de consola de comandes i amb permisos d'administrador, executarem les següents comandes:

```
# gunzip iperf-2.0.5.tar.gz → Extrau un arxiu iperf-2.0.5.tar
```

Descomprimeix els arxius continguts a iperf-2.0.5.tar

```
# tar xvf iperf-2.0.5.tar cd iperf-2.0.5
```

Execució del *script* de configuració del codi font per a la compilació

```
# ./configure
```

En aquest punt, el *script configure* indica que manquen determinades llibreries, per continuar amb el procés, s'ha d'instal·lar determinades llibreries de c++,

instal·lem aquelles necessàries utilitzant la següent comanda i continuem amb el procés:

```
# apt-get install g++ gawk
# ./configure
# Make
# Make install
```

Un instal·lada l'aplicació, podem fer una prova d'execució, mitjançant la sentència:

```
# iperf
```

sabent que obtindrem un error degut a no s'especifica cap servidor origen ni destí, però ens indica que el servei *iperf* ha estat configurat adequadament.

K.1.3 Apache Web Server

Per a la instal·lació del servei s'ha d'executar la comanda mostrada a continuació amb drets d'administrador:

```
# apt-get install apache2
```

Es pot comprovar si s'ha instal·lat correctament accedint des de un navegador web a l'adreça IP del servidor:

- <http://147.83.118.250/>

Es mostra una plana web indicant que: *It works!*

K.1.4 Bandwith Test controller

En aquest cas també cal descarregar l'eina en un paquet comprimit des de [23], es descarrega la versió 1.2a tal i com s'indica a la guia d'administració [9]. Per a la instal·lació del servei s'han d'executar les comandes indicades a continuació.

Descompressió de l'arxiu *.tar.gz*

```
# tar -xvf bwctl-1.2a.tar.gz
```

Canvi al nou directori

```
# cd bwctl-1.2a
```


Compilació, i instal·lació

```
# ./configure
# make
# make install
```

Un cop instal·lada l'aplicació s'han de modificar els arxius de configuració del servei que es troben al directori *'conf'* dins la ruta de instal·lació.

- Bwctld.conf

```
iperf_cmd      iperf
```

Indiquem al procés *bwctl* que *iperf* es pot executar directament al estar inclòs al PATH de l'equip.

```
iperf_port      5004
```

Aquest cop estem indicant que quan s'hagi d'invocar el procés *iperf*, s'indiqui que el port a utilitzar sigui el 5004, ja que trobem a partir de la ordre del sistema *'netstat'* que el port 5001, per defecte, és utilitzat pel servei Apache Tomcat.

```
root_fooly
```

A l'hora de descarregar el servei, s'observa que aquesta opció està comentada (amb el símbol # al davant), això faria que el servei *bwctld* no es pogués executar com a usuari root.

També cal realitzar modificacions en el fitxer *bwctld.limits* en el que s'especifiquen les subxarxes de les que s'acceptaran peticions, el límit de proves simultànies, o els recursos a assignar. A la nostra configuració afegim la subxarxa d'equips del laboratori com a subxarxa permesa.

- Bwctld.limits

Cal afegir la línia *'Assign net'* que correspongui de la manera indicada

```
# nmslan networks
Assign net 147.83.118.0/24 ami
```

Un cop configurats aquests arxius, ja es pot revisar el funcionament del servei utilitzant les següents comandes, la primera executa el servei *bwctld* i la segona, realitza una prova de connexió amb un servei *BWCTL* situat a Zagreb.

```
# bwctld -c ./conf
# bwctl -s 147.83.118.250 -c 193.198.229.166
```

Per a que un servidor pugui rebre peticions de proves, ha d'estar funcionant el servei *bwctld*. En el cas de que un equip no tingui arrencat el servei *bwctld* i aquest equip executi la segona comanda, el procés *bwctl* emularà el servei *bwctld* a l'equip local.

A continuació, un exemple dels missatges que apareixen per consola al executar una prova d'ample de banda:

```
bwctl: 17 seconds until test results available

RECEIVER START
bwctl: exec_line: iperf -B 193.198.229.166 -s -f b -m -p 5022 -t 10
bwctl: start_tool: 3496853345.512063
-----
Server listening on TCP port 5022
Binding to local address 193.198.229.166
TCP window size: 87380 Byte (default)
-----
[ 15] local 193.198.229.166 port 5022 connected with 147.83.118.250
port 5022
[ 15] 0.0-10.1 sec 98246656 Bytes 77712652 bits/sec
[ 15] MSS size 1448 bytes (MTU 1500 bytes, ethernet)
bwctl: stop_exec: 3496853359.771131

RECEIVER END
```

K.2 Instal·lació del servei

Durant la instal·lació d'aquest servei, es troba que determinats paquets dels que s'indiquen a la guia d'administració no són disponibles per a la seva descàrrega, com és el cas del paquet *oppd-WebAdmin*. Segons s'extreu a partir de comunicació amb la llista de correu [27], aquests paquets tenien dependències que no resolubles, de manera que es decideix instal·lar la versió 0.52 encara en desenvolupament.

Per a la instal·lació d'aquest servei s'ha d'accedir a la consola de comandes i executar les següents comandes amb permisos d'administrador de l'equip:

1. Eliminar la versió antiga (si existeix una instal·lació prèvia)

```
# apt-get remove --purge perfsonar-oppd
```

2. Eliminar els següents arxius (si existeix una instal·lació prèvia)

```
# rm -rf /usr/lib/perfsonar/services/oppd
# rm /etc/oppd.conf
```

3. Eliminar el paquet *NMVG*

```
# apt-get remove --purge libnmwg-perl
```

4. Verificar que l'arxiu */etc/apt/sources.list* conté la següent entrada

```
# deb http://downloads.perfsonar.eu/repositories/deb unstable main
```

5. Instal·lar el paquet 0.52 juntament amb les dependències.

```
# apt-get install perfsonar-oppd-mp-bwctl/unstable
```

Trobem a [29], que per a la versió 3.3 del perfSONAR MDM, aquest servei passarà a anomenar-se OPPD, *Open Perl PerfSONAR Daemon*, aquest és el motiu de la variació de la nomenclatura del paquet a instal·lar.

Un cop acabada la instal·lació, ens trobem els primers problemes, per tant es passa ara a l'apartat dels problemes de configuració.

K.3 Proves amb el servei BWCTL

S'han efectuat algunes mesures a serveis BWCTL de diferents pesos, a continuació es presenten els amplituds de banda obtinguts, per a veure si s'aprecien diferències depenent de la situació geogràfica del servidor de destí.

El primer cas mostrat és per a les proves realitzades amb servidors Internet 2, en aquest cas s'han efectuat proves entre el perfhos i el servidor en concret, obtenint la gràfica que es mostra a continuació.

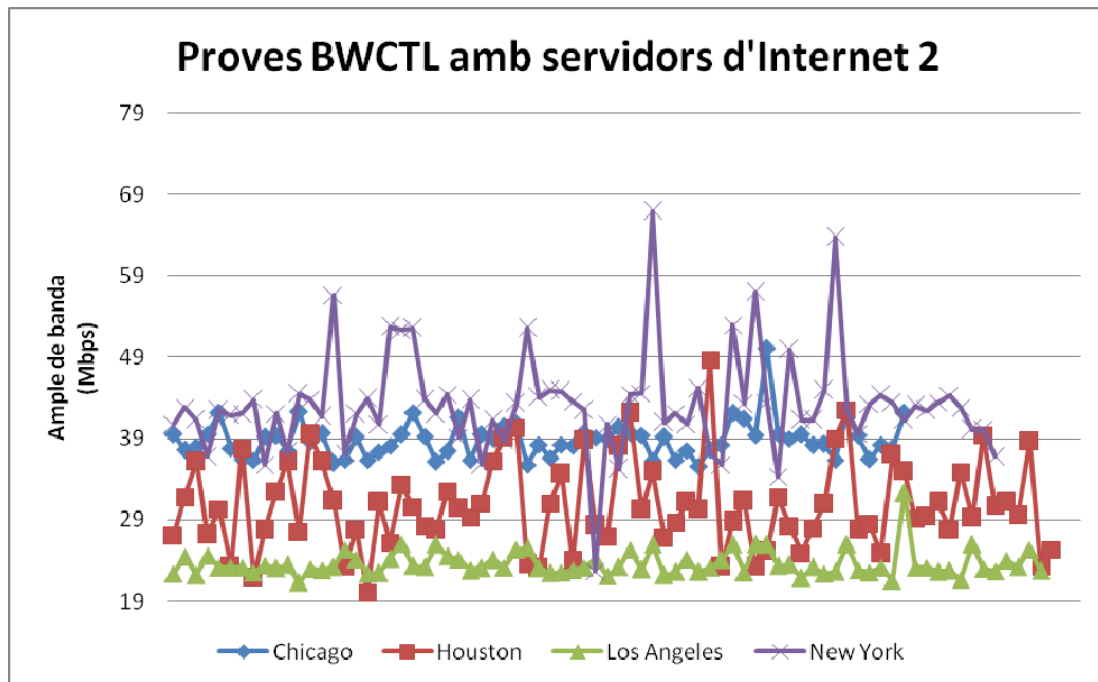


Fig K.1 Gràfica mostrant l'ample de banda obtingut per la prova BWCTL amb servidors d'estats units.

S'aprecia en aquesta gràfica que les proves als equips de la costa Est són millors que les realitzades a la costa Oest. Utilitzant l'eina tracert es determina el nombre de salts a cadascun dels servidors provats, es poden comparar els resultats en la taula a continuació:

Taula K.1 Proves amb l'eina BWCTL i servidors Internet2

Ciutat	Nombre de Salts	Valor mig d'ample de banda
Chicago	13	38.86 Mbps
Houston	14	30.84 Mbps
Los Angeles	15	23.63 Mbps
New York	13	43.23 Mbps

En el cas de les connexions amb servidors d'Estats Units és on més s'aprecia el factor de la distància, ja que tot i que segurament el camí que segueixen els paquets per arribar als Estats Units (entrada desde New York o Washington), s'aprecia que les proves amb els servidors de Houston i Los Angeles són les que obtenen pitjors resultats sent les ciutats més allunyades.

S'han seleccionat també quatre punts de la xarxa Geant2, en aquest cas les proves s'han efectuat en ambdós sentits de la comunicació, donat que els equips son a la xarxa geant, és possible que la transmissió segueixi diferents camins i la distancia no sigui un factor tan determinant com en el cas anterior en el que calia travessar un enllaç Géant2-Internet2.

Observem primer la gràfica de la connexió entre el servidor perfhost i els altres servidors.

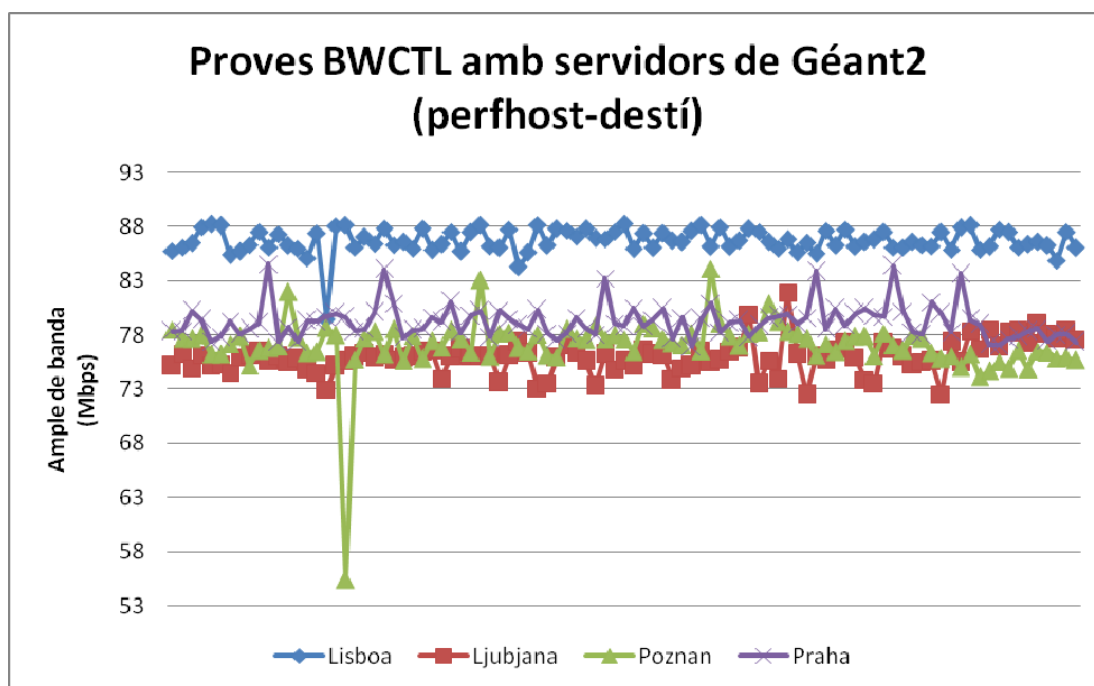


Fig. K.2 Resultats de les proves d'ample de banda BWCTL entre el perfhost i el destí indicat.

De la mateixa manera que a la prova anterior, es comparen a continuació el nombre de salts que ens indica l'eina tracert amb la mitjana de l'ample de banda.

Taula K.2. Proves amb l'eina BWCTL entre el perfHOST i servidors de Géant2

Ciutat	Nombre de Salts	Valor mig d'ample de banda
Lisboa	12	86.69 Mbps
Ljubjana	14	75.95 Mbps
Poznan	13	77.08 Mbps
Praha	13	79.25 Mbps

S'aprecia que el nombre de salts està directament relacionat amb el Valor mig de l'ample de banda.

Al camí contrari, definint que sigui el host remot qui iniciï la transmissió, s'obté la següent gràfica:

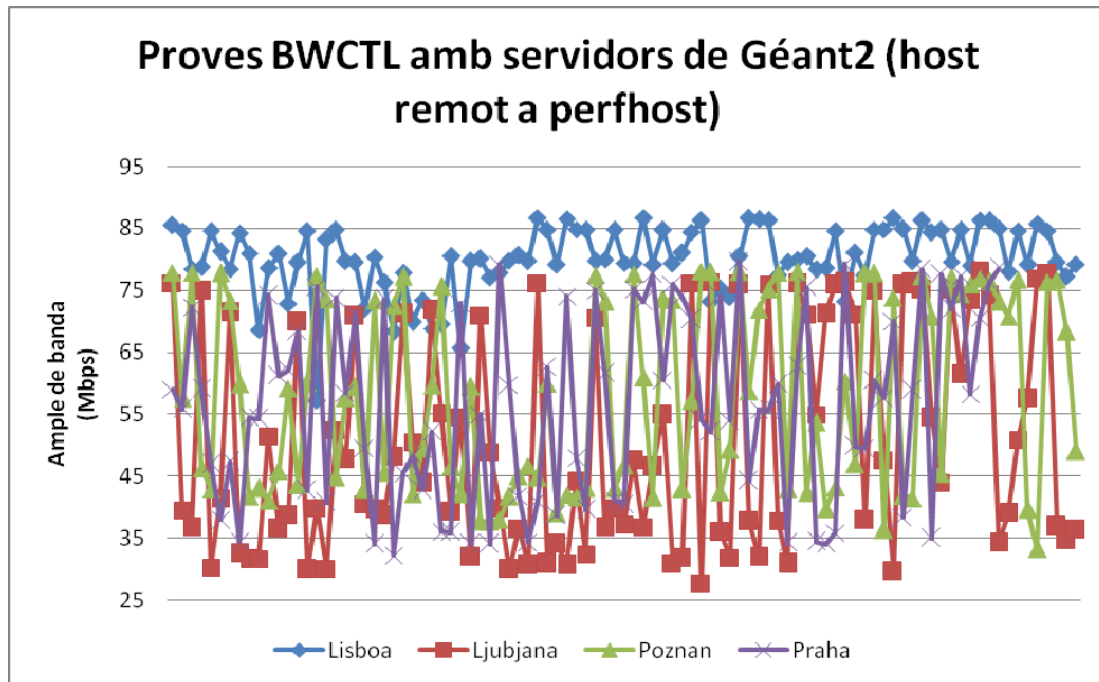


Fig K.3 Gràfica amb l'ample de banda assolit a les proves des de hosts remots cap al perfhost.

Cal apreciar en aquest punt la diferència que existeix entre proves efectuades des del mateix host. Aquí pot haver influït la xarxa del laboratori com a coll d'ampolla ja que aquests servidors disposaran segurament d'una gran capacitat de sortida, de manera que si detecten pèrdua de paquets la seva finestra de transmissió del protocol TCP es disminueixi, comportant aquests pitjors resultats. La relació nombre de salts i l'ample de banda mig, si que es segueix acomplint.

A continuació es mostra la taula corresponent a aquestes proves:

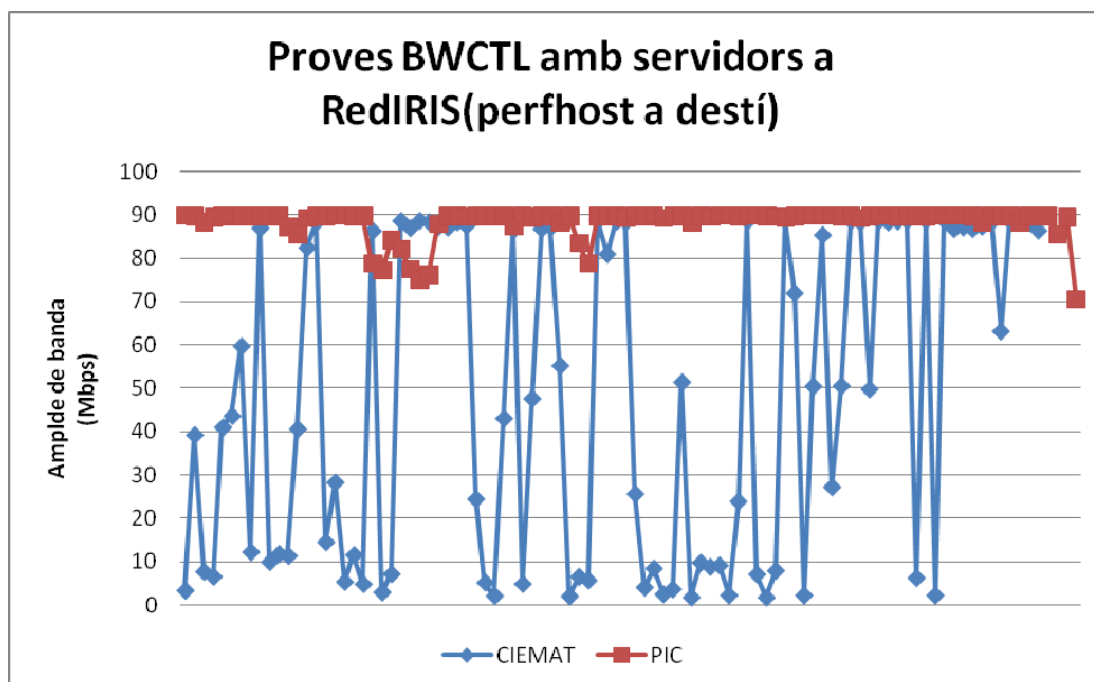
Taula K.3. Proves amb l'eina BWCTL entre servidors de Géant2 i perfHOST

Ciutat	Nombre de Salts	Valor mig d'ample de banda
Lisboa	12	80.00 Mbps
Ljubjana	14	50.90 Mbps
Poznan	13	58.40 Mbps
Praha	13	56.69 Mbps

En aquest cas s'aprecia una diferència notable entre els resultats obtinguts amb el servei de Lisboa enfront als altres serveis. Això pot ser degut a que la connexió a Lisboa és directa amb el node central de RedIRIS a Madrid, mentre que la connexió amb els altres països segurament passarà per més d'una xarxa nacional de recerca.

Les darreres proves han estat efectuades entre dos serveis *BWCTL* a RedIRIS. S'han escollit dos serveis BWCTL bastant propers geogràficament al perfhosht. El primer punt és situat al CIEMAT, el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, que té la seu central a Madrid i també té seu a Barcelona. I la segona és el PIC, el Port d'Informació Científica, que és l'únic TIER-1 del LHC a l'estat Espanyol i està situat al campus de la Universitat Autònoma de Bellaterra.

De la mateixa manera que les proves contra equips de la xarxa Géant2, es mostra primer el resultat de les proves establint la connexió des de el perfhosht.

**Fig K.4** Resultats de les proves d'ample de banda contra servidors a RedIRIS

Trobem que tot i la proximitat dels equips, suposem el primer a Madrid i el segon a Barcelona, els resultats són diferents en el cas de l'equip de Madrid. No podem determinar amb l'eina *tracert* el nombre de salts que hi ha cap als equips, ho intentem fer apuntant a algun dels serveis web, però no és possible de determinar el nombre de salts per aquests centres.

Taula K.4. Proves amb l'eina BWCTL entre *perfHOST* i servidors a RedIRIS

Centre	Nombre de Salts	Valor mig d'ample de banda
CIEMAT	Desconegut	48.44 Mbps
PIC	Desconegut	88.27 Mbps

De la mateixa manera com a la prova anterior, observem a continuació el resultat a la inversa, sent el *sendhost* el servei remot.

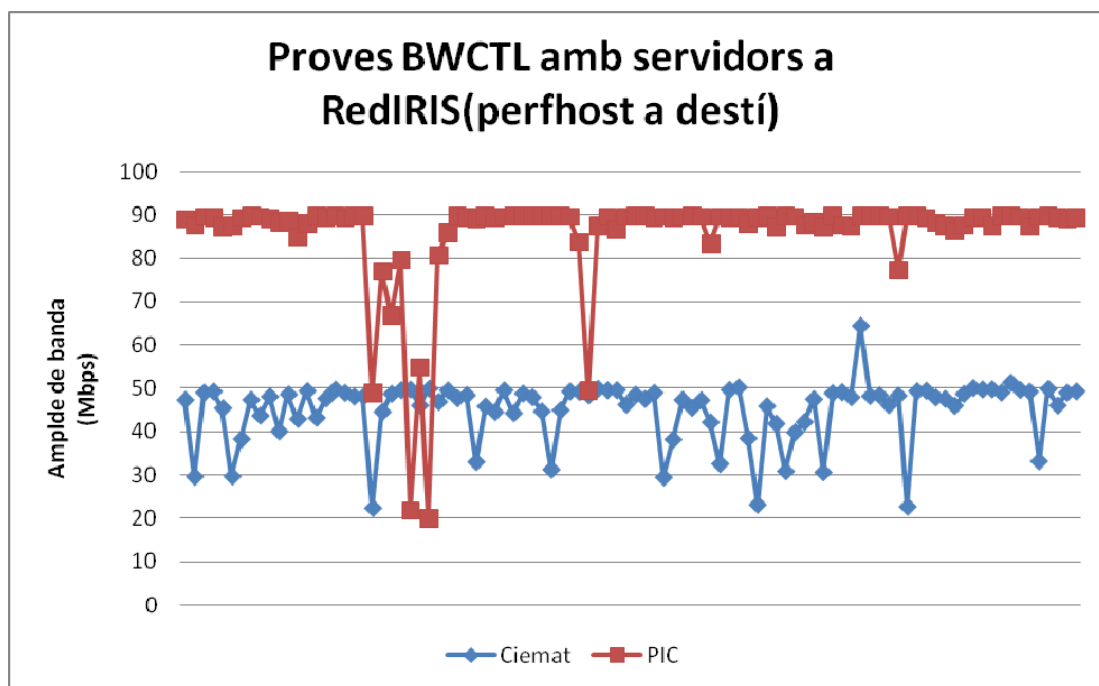


Fig K.5 Gràfica que mostra els amplex de banda assolits a proves BWCTL entre servidors a RedIRIS i el *perfHOST*

Es pot apreciar a la taula que amb el servei *bwctl* del PIC, a Bellaterra, obté molt millors resultats que amb el servei del centre Ciemat, que com es comentava se suposa que està a Madrid.

A la taula a continuació mostrem també el promig de les mesures.

Taula K.5. Proves entre servidors de RedIRIS i perfHOST

Centre	Nombre de Salts	Valor mig d'ample de banda
CIEMAT	Desconegut	45.09 Mbps
PIC	Desconegut	85.57 Mbps

ANNEX L.INSTAL·LACIÓ I CONFIGURACIÓ DEL SERVEI TELNET/SSH

L.1 Instal·lació

La instal·lació del servei requereix de l'execució de la següent comanda amb permisos d'administrador:

```
# apt-get install perfsonar-java-sshtelnet-mp
```

La instal·lació acaba reiniciant el servei Tomcat.

L.2 Configuració bàsica del servei

Un cop es tenen els arxius necessaris, és el moment de realitzar la configuració mínima del servei, es descriuen les opcions tal i com apareixen a la versió 1.3.4.1 del servei. Cal accedir a la plana de configuració del servei a través de la URL :

<http://147.83.118.250:8180/ps-mdm-sshtelnet-mp>

A continuació cal entrar a l'apartat 'Service' del menú 'Basic Configuration', on es requereix indicar usuari i *password* per defecte d'entrada al servei, 'perfsonaruser' i 'perfsonarpass' respectivament.

La pantalla de configuració és molt semblant a les d'altres serveis, s'indiquen a continuació els apartats més característics:

- *Enter the name of the metadata configuration file:*
 - El servei demana el nom de l'arxiu de metadades, tal com s'ha indicat abans, per simplificar la configuració, es recomanable utilitzar el valor per defecte.
- *Enter the full path of the metadata configuration file:*
 - S'ha d'indicar aquí la ruta completa de l'arxiu de configuració, nom de l'arxiu inclòs.

L.3 Configuració del servei

Donat que es requereixen determinats dispositius de xarxa compatibles per a la implementació del servei, amb aquest servei no s'han pogut realitzar proves de configuració i de funcionament com amb d'altres serveis. En aquest apartat doncs es comenten els arxius necessaris per a la configuració del servei.

Per a un correcte funcionament i comunicació entre el servei i els dispositius de xarxa a definir, cal crear dos arxius de configuració.

Arxiu de configuració de Metadades

Per a l'execució de comandes *SHOW* als *routers* que es donin d'alta al servei, s'han de definir el tipus de comanda a executar.

La definició es fa a partir d'un arxiu XML de metadades, donat que cada model de dispositiu pot tenir les seves pròpies comandes. El servei inclou dos arxius que indiquen les comandes compatibles. Els arxius es poden consultar a la següent ruta:

```
# /usr/lib/perfsonar/services/ps-mdm-sshtelnet-mp/WEB-INF/doc/commands/commands-complete-list-v6-AI.xls

# /usr/lib/perfsonar/services/ps-mdm-sshtelnet-mp/WEB-INF/doc/commands/commands-minimum-list-v6-SM.xls
```

També es proporcionen alguns *scripts* per a la generació dels arxius de metadades, que es poden trobar a la ruta:

```
# /usr/lib/perfsonar/services/ps-mdm-sshtelnet-mp/WEB-INF/classes/perfsonar/contrib/metadata_configuration_tools/
```

Aquestes eines ens permetran la creació de:

- Arxiu CSV de *routers* i comandes a donar d'alta al servei. Especialment indicat per si es vol configurar un nombre elevat de dispositius.
- Eina de configuració, que permet la creació dels arxius de metadades. Per a generar aquest arxiu es requereix proporcionar la llista CSV de comandes i *routers*.

Es pot trobar un arxiu de metadades de mostra a la ruta a continuació:

```
# /usr/lib/perfsonar/services/ps-mdm-sshtelnet-mp/WEB-INF/doc/Metadata Configuration file for SSHTelnet 1.3.doc
```

Cal indicar que la configuració que es mostra és específica per a equipament *Cisco* i que els passwords dels dispositius, que s'indiquen als arxius de configuració, han d'ésser codificats en *BASE64*.

ANNEX M. ARXIUS DE CONFIGURACIO DE SERVEIS A LA INTERFÍCIE GRÀFICA

Durant la realització de proves s'ha d'indicar a la interfície perfSONAR UI els paràmetres dels serveis contra els que es volen realitzar aquestes proves, un cop es selecciona la opció *Select Service Addresses* apareix una finestra com la que es mostra a la imatge a continuació.

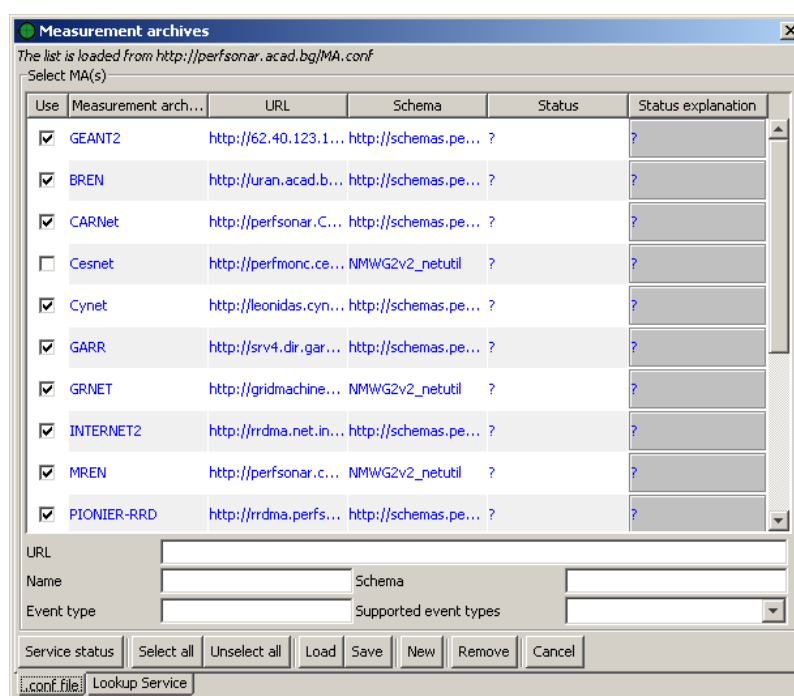


Fig L.1 Pantalla de selecció del servei perfSONAR a consultar

Si fem click sobre d'algun dels serveis, es permet modificar els paràmetres del mateix a la part inferior, de manera que es poden definir els serveis propis indicant la URL d'entrada al servei, el esquema utilitzat, el paràmetre de tipus d'event, l'esquema i el tipus de events permesos.

D'aquesta manera es pot utilitzar la opció *Save* per a desar aquelles modificacions que s'hagin efectuat i posteriorment carregar-les a la pantalla a través de la opció *Load*. D'aquesta manera podrem tenir un arxiu amb els serveis que utilitzem més sovint.

ANNEX N. IMPLEMENTACIÓ MINIMA DE LES CLASSES PerfsonarTab I MainComponent

A continuació es presenta un codi “Hola Mon” integrat a la consola perfSONAR UI.

- Classe PestanyaPerfsonarTab

```
package org.perfsonar.perfsonarui.pestanyablanc;

import java.awt.Component;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import org.perfsonar.perfsonarui.IPerfsonarSchema;
import org.perfsonar.perfsonarui.PerfsonarSchema;
import org.perfsonar.perfsonarui.SmartMARequest;
import org.perfsonar.perfsonarui.ma.MARequest;

import org.perfsonar.perfsonarui.plugins.IPerfsonarRequest;
import org.perfsonar.perfsonarui.plugins.MAEndpoint;
import org.perfsonar.perfsonarui.ui.PerfsonarModel;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;
import org.perfsonar.perfsonarui.ui.actions.PerfsonarRequestAction;
import org.perfsonar.perfsonarui.util.Collections;

public class PestanyaPerfsonarTab extends PerfsonarModel {

    protected int endpointSrcIndex = -1;

    protected int endpointDstIndex = -1;

    private PestanyaMainComponent panel;

    public PluginMainPanel createMainComponent(Component
parentComponent) {

        panel = new PestanyaMainComponent(parentComponent, this);
        return panel;
    }

    public int getOrder() {
        return 4;
    }

    public String toString()
    {
        return "Hola Mon!";
    }

    @Override
```

```

    protected SmartMARequest createSmartRequest() {
        if (request != null) {
            return request;
        }

        return new SmartMARequest() {
            public IPerfsonarRequest createMARequest(MAEndpoint
            endpoint) {
                return new MARequest();
            }
        };
    }

    @Override
    public List<IPerfsonarSchema> getSupportedRequests() {
        ArrayList<IPerfsonarSchema> supported = new
        ArrayList<IPerfsonarSchema>();
        supported.add(PerfsonarSchema.getInstance("Hello World"));
        return supported;
    }

    @Override
    public Set<String> getEventTypes() {
        Set<String> eventTypes = new HashSet<String>();
        return Collections.unmodifiableCopy(eventTypes);
    }

    @Override
    protected PerfsonarRequestAction createMetadataRequestAction() {
        return null;
    }
}

```

Sense la classe que s'ha mostrat anteriorment, el sistema de càrrega de *plugins* de la perfSONAR UI no seria capaç de carregar la pestanya i el panell que es carrega des de la classe PestanyaMainComponent i que es mostra a continuació.

Les pestanyes MainComponent importen diverses llibreries de javax.swing, molt utilitzades per a la realització de programes Java amb interfície gràfica.

- Classe PestanyaMainComponent

```

package org.perfsonar.perfsonarui.pestanyablanc;

import java.awt.Component;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import javax.swing.BorderFactory;
import javax.swing.JTextArea;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
//import Boto
import javax.swing.JButton;

import org.perfsonar.perfsonarui.plugins.IPerfsonarTab;
import org.perfsonar.perfsonarui.plugins.MAEndpoint;
import org.perfsonar.perfsonarui.plugins.MAEndpointList;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;

/**
 * Pestanya visualization for perfSonarUI
 */

public class PestanyaMainComponent extends PluginMainPanel
    implements ActionListener
{
    private static final long serialVersionUID = 1016070292100053577L;

    float numseg;
    int cont=0;

    protected JTextArea result;
    protected JButton boto;

    public PestanyaMainComponent(Component parentComponent,
        IPerfsonarTab psModel) {
        super(parentComponent, psModel, new GridLayout());
    }

    public void addWidgets() {
        JPanel panel = new JPanel();
        GridBagConstraints constraints;
        this.setLayout(new GridLayout());
        panel.setLayout(new GridBagLayout());
        constraints = new GridBagConstraints();
        constraints.fill = GridBagConstraints.BOTH;
        constraints.insets = new Insets(2,2,2,2);

        boto = new JButton("Prova");
        constraints.gridx +=15;
        boto.setVisible(true);
        panel.add(boto, constraints);
        boto.setActionCommand("hola");
        boto.addActionListener(this);

        result = new JTextArea("");
        result.setBorder(BorderFactory.createEtchedBorder());
    }

```



```
        constraints.gridx = 0;
        constraints.gridy = 14;
        constraints.gridwidth = 5;
        constraints.fill = GridBagConstraints.BOTH;
        result.setPreferredSize(new Dimension(200,200));

        JScrollPane sp = new JScrollPane(result);
        sp.setBorder(BorderFactory.createTitledBorder("Result"));
        result.setWrapStyleWord(true);
        sp.setPreferredSize(new Dimension(200,200));
        panel.add(sp, constraints);

        JScrollPane mainSrcPane = new JScrollPane(panel);
        add(mainSrcPane);
    }

    public void actionPerformed(ActionEvent event)
    {
        String so;

        result.insert("HOLA MON!!!", 0);

    }
}
```

ANNEX O. CODI DE LA PESTANYA MOSTRATEXT

A continuació es pot trobar el codi utilitzat per la pestanya MostraText

- Classe MostraTextPerfsonarTab

```
package org.perfsonar.perfsonarui.MostraText;

import java.awt.Component;
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import javax.swing.Action;
import javax.swing.JDialog;
import org.perfsonar.perfsonarui.IPerfsonarSchema;
import org.perfsonar.perfsonarui.PerfsonarSchema;
import org.perfsonar.perfsonarui.SmartMARequest;
import org.perfsonar.perfsonarui.ma.MARequest;
import org.perfsonar.perfsonarui.plugins.IPerfsonarRequest;
import org.perfsonar.perfsonarui.plugins.MAEndpoint;
import org.perfsonar.perfsonarui.plugins.MAEndpointList;
import org.perfsonar.perfsonarui.ui.PerfsonarModel;
import org.perfsonar.perfsonarui.ui.actions.PSActionMap;
import org.perfsonar.perfsonarui.ui.actions.PSUIAction;
import org.perfsonar.perfsonarui.ui.actions.PerfsonarRequestAction;
import org.perfsonar.perfsonarui.ui.actions.ShowEndpointsAction;
import org.perfsonar.perfsonarui.util.Collections;

/**
 *
 * MostraText Panel, panell que permet escriure un text per a capturar-
lo i mostrar-lo
 * per pantalla a un quadre de text no editable i a una infoline
 */
public class MostraTextPerfsonarTab extends PerfsonarModel {

    protected MAEndpointList destinations;

    protected int endpointSrcIndex = -1;

    protected int endpointDstIndex = -1;

    private MostraTextMainComponent pane;

    protected PSActionMap accions;

    public MostraTextPerfsonarTab()
    {
        this(null);
    }
}
```

```

public MostraTextPerfsonarTab(MAEndpointList endpoints)
{
    super(endpoints);

    for (int i=0; i < endpointList.size();i++)
    {
        if (endpointList.get(i).isEnabled())
        {
            userData.setEndpoint(endpointList.get(i));
            break;
        }
    }

    //actions
    accions = new PSActionMap();

    endpointsAction =
createEndpointsAction(getEventTypes(),getEndpointList());
    accions.put(endpointsAction);
    endpointsAction.setActions(accions);

    clearAction = createClearAction();
    accions.put(clearAction.getValue(Action.NAME),clearAction);
    clearAction.setActions(accions);
}
/**
 *
 */
protected SmartMARequest createSmartRequest()
{
    if (request != null)
    {
        return request;
    }

    return new SmartMARequest() {
        public IPerfsonarRequest createMARequest(MAEndpoint
endpoint) {
            return new MARequest();
        }
    };
}

public int getOrder()
{
    return 5;
}

public MostraTextMainComponent createMainComponent(Component
parentComponent)
{
    pane = new MostraTextMainComponent(parentComponent, this);
    return pane;
}

public PerfsonarRequestAction createMetadataRequestAction(){
    return null;
}

```

```

    }

    //TITOL DE LA PESTANYA
    public String toString()
    {
        return "Mostra Text";
    }

    public List<IPerfsonarSchema> getSupportedRequests()
    {
        ArrayList<IPerfsonarSchema> supported;

        supported = new ArrayList<IPerfsonarSchema>();
        supported.add(PerfsonarSchema.getInstance("bwctl"));
        return supported;
    }

    /**
     * BWCTL can only deal with one MP, set the appropriate flag for
the
     * endpoint list.
     */
    public MAEndpointList createEndpoints()
    {
        MAEndpointList list;
        list = super.createEndpoints();
        list.setMultipleEndpoints(false);
        return list;
    }

    /**
     * Override in order not to create a default Options button.
     */
    public PSUIAction createOptionsAction() {
        ShowEndpointsAction action = new
ShowEndpointsAction(getEventTypes(), getDestinations(), "BWCTL
Destination"){
            protected JDialog createDialog(Set<String>
eventTypes){
                setEndpointSelected(endpoints,
endpointDstIndex);
                return super.createDialog(eventTypes);
            }
        };

        action.addListener(new ShowEndpointsAction.Listener() {

            public void selectionChanged(Collection<MAEndpoint>
selected) {

                //pane.updateDestination();
                //mètode eliminat de la classe
                //MostraTextMaincomponent
            }
        });
        return action;
    }

```

```

    public PSUIAction createEndpointsAction(Set<String> eventTypes,
MAEndpointList endpointList) {
        ShowEndpointsAction action = new
ShowEndpointsAction(eventTypes, endpointList, "BWCTL MP/Source"){

            protected JDialog createDialog(Set<String>
eventTypes){
                setEndpointSelected(endpoints,
endpointSrcIndex);
                return super.createDialog(eventTypes);
            }
        };

        action.addListener(new ShowEndpointsAction.Listener() {

            public void selectionChanged(Collection<MAEndpoint>
selected) {

                //pane.updateSource();
                //méthode éliminat de la classe
                //MostraTextMaincomponent
            }
        });
        return action;
    }

    public MAEndpointList getDestinations()
    {
        if (destinations == null)
        {
            destinations = getEndpointList();

            /* Disable them all */
            for (int i = 0; i < destinations.size(); i++)
            {
                destinations.get(i).setEnabled(false);
            }
        }
        return destinations;
    }

    public void setEndpointSelected(MAEndpointList endpoints, int
index){
        for(int i = 0; i < endpoints.size(); i++){
            if(i!=index)endpoints.get(i).setEnabled(false);
            else endpoints.get(i).setEnabled(true);
        }
    }

    @Override
    public Set<String> getEventTypes() {
        Set<String> eventTypes = new HashSet<String>();
        //TODO eventTypes.add("");
        return Collections.unmodifiableCopy(eventTypes);
    }

    @Override
    public PSActionMap getActions() {

```

```

        return accions;
    }
}

```

- Classe MostraTextMainComponent

```

package org.perfsonar.perfsonarui.MostraText;

import java.awt.Component;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.net.URL;
import java.util.Hashtable;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

import org.perfsonar.perfsonarui.plugins.IPerfsonarTab;
import org.perfsonar.perfsonarui.plugins.MAEndpoint;
import org.perfsonar.perfsonarui.plugins.MAEndpointList;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;

/**
 *
 * MostraText Panel, panell que permet escriure un text per a capturar-
lo i mostrar-lo
 * per pantalla a un quadre de text no editable i a una infoline
 */

public class MostraTextMainComponent extends PluginMainPanel
    implements ActionListener
{
    private static final long serialVersionUID = 1016070292100053577L;

    float numseg;
    int cont=0;
    protected Hashtable<String,String> namePairs = new
Hashtable<String,String>();

    protected JLabel infoLabelLine2;
    protected JTextArea resultat;
    protected JButton boto;
    protected JTextArea prova;

    String res;

```

```

    public MostraTextMainComponent(Component parentComponent,
        IPerfsonarTab psModel) {
        super(parentComponent, psModel, new GridLayout());
    }

    public void addWidgets() {
        JPanel panel = new JPanel();
        GridBagConstraints constraints;
        this.setLayout(new GridLayout());
        panel.setLayout(new GridBagLayout());
        constraints = new GridBagConstraints();

        constraints.gridwidth = 1;
        constraints.gridheight = 4;

        //AREA DE TEXT
        prova = new JTextArea("");
        prova.setBorder(BorderFactory.createEtchedBorder());

        //LIMIT DeL PANELL, EN EL QUE S'INCLOU EL PANELL
        JScrollPane scrollpanel = new JScrollPane(prova);
        scrollpanel.setBorder(BorderFactory.createTitledBorder("Escriu al Cuadre"));
        prova.setWrapStyleWord(true);
        scrollpanel.setPreferredSize(new Dimension(500,200));
        constraints.fill= GridBagConstraints.HORIZONTAL;
        constraints.gridx = 0;
        constraints.gridy = 0;
        panel.add(scrollpanel, constraints);
        boto = new JButton("Prova");
        boto.setVisible(true);
        constraints.gridx = 0;
        constraints.gridy = 5;
        panel.add(boto, constraints);
        boto.addActionListener(this);

        resultat = new JTextArea("");
        resultat.setBorder(BorderFactory.createEtchedBorder());

        resultat.setEditable(false);
        JScrollPane sp = new JScrollPane(resultat);
        sp.setBorder(BorderFactory.createTitledBorder("Resultat"));
        resultat.setWrapStyleWord(true);
        constraints.fill= GridBagConstraints.HORIZONTAL;
        constraints.gridx = 0;
        constraints.gridy = 10;
        panel.add(sp, constraints);

        infoLabelLine2 = new JLabel(" ");
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.gridx = 0;
        constraints.gridy = 15;

        panel.add(infoLabelLine2, constraints);
        infoLabelLine2.setText(" ");
    }

```

```
        JScrollPane mainSrcPane = new JScrollPane(panel);
        add(mainSrcPane);
    }

    public String getEndpointHost(MAEndpointList endpoints)
    {
        if (endpoints == null)
        {
            return null;
        }

        for (int i = 0; i < endpoints.size(); i++)
        {
            MAEndpoint endpoint;
            URL address;

            endpoint = endpoints.get(i);
            if (! endpoint.isEnabled())
            {
                continue;
            }

            try
            {
                address = new URL(endpoint.getEndpoint());
            }
            catch (java.net.MalformedURLException e)
            {
                logger.error("Malformed URL: " + e);
                return null;
            }
            /* Needs a bit more work for IPv6
             * service addresses, see RFC 2732.
             */
            return address.getHost();
        }

        /* No active endpoint */
        return null;
    }

    public String getEndpointName(MAEndpointList endpoints)
    {
        if (endpoints == null)
        {
            return null;
        }

        for (int i = 0; i < endpoints.size(); i++)
        {
            MAEndpoint endpoint;
            String name;

            endpoint = endpoints.get(i);
            if (! endpoint.isEnabled())
```



```
        {
            continue;
        }

        name = endpoint.getTitle();

        /* Needs a bit more work for IPv6
         * service addresses, see RFC 2732.
         */
        return name;
    }

    /* No active endpoint */
    return null;
}

public void actionPerformed(ActionEvent event)
{
    String so;
    so=prova.getText();
    resultat.setText(so);
    infoLabelLine2.setText(so);
}

public void clearJText(){
    resultat.setText(" ");
}

}
```

ANNEX P. CODI DE LA PESTANYA PINGERSQL

Aquesta pestanya permet a l'usuari seleccionar un host al que fer-hi pings, aïllar el resultat del RTT i emmagatzemar aquest valor a una base de dades SQL creada per a la realització d'aquest treball.

Donat que la classe `PingerSqlPerfsonarTab` és molt semblant a la classe `MostraTextPerfsonarTab`, donat que aquestes classes són les que gestionen les pestanyes, es mostra tan sols la classe `PingerSqlMainComponent`, que és la que mostra l'aspecte que tindrà aquesta pestanya.

- Classe `PingerSqlMainComponent`

```
package org.perfsonar.perfsonarui.pingersql;

import java.awt.Component;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.Hashtable;
import java.util.List;

import javax.swing.BorderFactory;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

//import Boto
import javax.swing.JButton;

import org.perfsonar.perfsonarui.pingersql.PingerSqlGraphPanel;
import org.perfsonar.perfsonarui.pingersql.eseculer;
import org.perfsonar.perfsonarui.plugins.IPerfsonarTab;
import org.perfsonar.perfsonarui.plugins.MAEndpoint;
import org.perfsonar.perfsonarui.plugins.MAEndpointList;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;

/**
 * Pinger SQL fa pings a través del ping del sistema al host que
 * s'indiqui al quadre i desa
 * el RTT a una base de dades SQL
 */
```

```

public class PingerSqlMainComponent extends PluginMainPanel
    implements ActionListener
{
    private static final long serialVersionUID = 1016070292100053577L;

    float numseg;
    int cont=0;
    protected Hashtable<String,String> namePairs = new
Hashtable<String,String>();

    protected JTextArea destinationAddress;
    protected JLabel infoLabelLine2;
    protected JTextArea result;
    protected JButton boto;
    protected PingerSqlGraphPanel graph;
    String resultat;

    public PingerSqlMainComponent(Component parentComponent,
IPerfsonarTab psModel) {
        super(parentComponent, psModel, new GridLayout());
    }

    public void addWidgets() {
        JPanel panel = new JPanel();
        GridBagConstraints constraints;
        this.setLayout(new GridLayout());
        panel.setLayout(new GridBagLayout());
        constraints = new GridBagConstraints();
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.insets = new Insets(2,2,2,2);

        constraints.gridwidth = 1;
        constraints.gridheight = 4;

        destinationAddress = new JTextArea("");

        destinationAddress.setBorder(BorderFactory.createEtchedBorder());

        destinationAddress.setEditable(true);
        constraints.gridx = 0;
        constraints.gridy = 0;
        constraints.gridwidth = 5;

        JScrollPane scroll = new JScrollPane(destinationAddress);

        scroll.setBorder(BorderFactory.createTitledBorder("Escriu
l'adreça Ip a la que vols fer ping"));

        destinationAddress.setWrapStyleWord(true);
        panel.add(scroll, constraints);
    }
}

```

```

        boto = new JButton("Prova");
        boto.setVisible(true);
        constraints.gridx = 0;
        constraints.gridy = 5;
        panel.add(boto, constraints);
        boto.addActionListener(this);

        infoLabelLine2 = new JLabel(" ");
        constraints.fill = GridBagConstraints.HORIZONTAL;
        constraints.gridx = 0;
        constraints.gridy = 12;

        panel.add(infoLabelLine2, constraints);
        infoLabelLine2.setText(" ");

        graph = new PingerSqlGraphPanel(this);
        constraints.gridx = 0;
        constraints.gridy = 10;

        constraints.fill = GridBagConstraints.BOTH;

        panel.add(graph, constraints);

        result = new JTextArea("");
        result.setEditable(false);
        result.setBorder(BorderFactory.createEtchedBorder());
        JScrollPane sp = new JScrollPane(result);
        sp.setBorder(BorderFactory.createTitledBorder("Resultat"));
        result.setWrapStyleWord(true);

        constraints.gridx = 0;
        constraints.gridy = 15;
        constraints.gridwidth = 5;
        constraints.fill= GridBagConstraints.BOTH;
        panel.add(sp, constraints);

        JScrollPane mainSrcPane = new JScrollPane(panel);
        add(mainSrcPane);
    }

    public void actionPerformed(ActionEvent event)
    {
        String so;

        if(result.getText().contains("RTT")){
            clearJText();
            clearGraph();
        }

        List<String> commands = new ArrayList<String>();
        commands.add("ping");
    }

```

```
so=System.getProperty("os.name");

if (so.contains("Linux")){
    //El ping de unix no té per defecte un nombre de
    //pings, és infinit
    commands.add("-c");
    commands.add("5");
}

commands.add(destinationAddress.getText());

try {

so=doCommand(commands,destinationAddress.getText());

    } catch (IOException e) {
        so="No ha funcionat";
    }

    result.insert(so, 0);

}
public void clearInfoLabels(){

    infoLabelLine2.setText(" ");

}

public void addValueToGraph(double time, double value){
    graph.setChartValue(time, value);
}

public void addAverageToGraph(double startTime, double endTime,
double value){
    graph.setAverageValue(startTime, endTime, value);
}

public void clearGraph(){
    graph.clearChart();
}

public void clearJText(){
    result.setText("");
}

public String getEndpointName(MAEndpointList endpoints)
{
    if (endpoints == null)
    {
        return null;
    }
}
```

```

        for (int i = 0; i < endpoints.size(); i++)
        {
            MAEndpoint endpoint;
            String name;

            endpoint = endpoints.get(i);
            if (! endpoint.isEnabled())
            {
                continue;
            }

            name = endpoint.getTitle();

            /* Needs a bit more work for IPv6
             * service addresses, see RFC 2732.
             */
            return name;
        }

        /* No active endpoint */
        return null;
    }

    public String doCommand(List<String> command, String IP_dest)
        throws IOException
    {
        resultat="";
        String s = null, InsertQ;
        int inicit, finalt;
        float suma=0;
        char[] aux;
        String temps;
        boolean ok;

        ProcessBuilder pb = new ProcessBuilder(command);
        Process process = pb.start();

        BufferedReader stdInput      = new      BufferedReader(new
InputStreamReader(process.getInputStream()));
        BufferedReader stdError      = new      BufferedReader(new
InputStreamReader(process.getErrorStream()));

        // Lectura de la sortida del ping
        System.out.println("Aquí es mostra la sortida de la comanda
ping\n");
        cont=1;
        String rejectlines="";

        Float [] vector=null;
        esecueler a = new esecueler();

        ok=a.Connect();

        if(!ok){
            System.out.println("SQL Connection FAIL!");

```

```

        resultat="SQL Connection Fail";
        return resultat;
    }
    while ((s = stdInput.readLine()) != null)
    {
        temps="";
        aux=s.toCharArray();

        if (s.contains("tiempo")){
            //español, windows
            inicit=s.lastIndexOf("tiempo=");
            finalt=s.lastIndexOf("ms ");
            temps=temps.copyValueOf(aux, inicit+7, (finalt-
            (inicit+7)));
            numseg = Integer.parseInt(temps);
            cont++;
        }else if (s.contains("time")&&s.contains("icmp_seq")){
            //english, unix
            inicit=s.lastIndexOf("time=");
            finalt=s.lastIndexOf(" ms")-1;
            temps=temps.copyValueOf(aux, inicit, (finalt-
            inicit)+1);

            numseg = Float.parseFloat(temps);
            cont++;
        }else if
        (s.contains("time")&&s.contains("Reply")&&(!s.equals(rejectl
        ines))){
            //english, Windows
            inicit=s.lastIndexOf("time=")+5;
            finalt=s.lastIndexOf("ms ");
            temps=temps.copyValueOf(aux, inicit, (finalt-inicit));
            System.out.println(temps);
            numseg = Float.parseFloat(temps);

            suma=suma+numseg;
            resultat=resultat+"\n"+s;
            graph.setChartValue(cont, numseg);
            Calendar c = GregorianCalendar.getInstance();

            c.get(Calendar.HOUR);

            InsertQ="Insert into ping (data, hora, host, rtt)
            values(\""+c.get(Calendar.YEAR)+"-"+(c.get(Calendar.MONTH)+1)+"-
            "+c.get(Calendar.DATE)+

            "\",\""+c.get(Calendar.HOUR)+":"+c.get(Calendar.MINUTE)+":"+c.get(
            Calendar.SECOND)+"\", \""+IP_dest+"\", \""+numseg+"\"";

            System.out.println(InsertQ);

            a.Insert(InsertQ);

            cont++;
        }
    }

```

```
    }  
    graph.setAverageValue(1, cont-1 , suma/(cont-1));  
    return resultat;  
  }  
}
```

L'objecte gràfic s'extrau de la classe BWCTLGraphPanel, que proporciona l'objecte del panell i que s'ha adaptat correctament en aquest codi.

ANNEX Q. CODI DE LA PESTANYA SQL RETRIEVER

La pestanya SqlRetriever, s'encarrega de realitzar una consulta SQL a la taula on s'emmagatzemen els valors RTT obtinguts amb la pestanya PingerSql.

```
package org.perfsonar.perfsonarui.SqlSenderRetriever;

import java.awt.Component;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Hashtable;
import javax.swing.BorderFactory;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JSeparator;
import javax.swing.JTextArea;

//import Boto
import javax.swing.JButton;
import org.perfsonar.perfsonarui.SqlSenderRetriever.esecueler;
import org.perfsonar.perfsonarui.plugins.IPerfsonarTab;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;

/**
 * SqlServerRetriever visualization for perfSonarUI
 */

public class SqlSenderRetrieverMainComponent extends PluginMainPanel
    implements ActionListener
{
    private static final long serialVersionUID = 1016070292100053577L;

    float numseg;
    int cont=0;
    protected Hashtable<String,String> namePairs = new
    Hashtable<String,String>();

    protected JTextArea destinationAddress;
    protected JLabel infoLabelLine2;

    protected JSeparator resultSeparator;

    protected JTextArea resultat;
```

```

protected JButton boto;
protected JTextArea prova;
protected esecueler con;

protected SqlSenderRetrieverGraphPanel graph;

String res, ip, query;
ResultSet response;
JScrollPane sp;

public SqlSenderRetrieverMainComponent(Component parentComponent,
IPerfsonarTab psModel) {
    super(parentComponent, psModel, new GridLayout());
}

public void addWidgets() {

    JPanel panel = new JPanel();
    GridBagConstraints constraints;
    this.setLayout(new GridLayout());
    panel.setLayout(new GridBagLayout());
    constraints = new GridBagConstraints();
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.insets = new Insets(2,2,2,2);

    constraints.gridwidth = 1;
    constraints.gridheight = 4;

    //AREA DE TEXT

    destinationAddress = new JTextArea("");
    destinationAddress.setBorder(BorderFactory.createEtchedBorder());
    destinationAddress.setEditable(true);
    constraints.gridx = 0;
    constraints.gridy = 0;
    constraints.gridwidth = 5;

    JScrollPane scroll = new JScrollPane(destinationAddress);
    scroll.setBorder(BorderFactory.createTitledBorder("Escriu l'adreça Ip a Consultar"));
    destinationAddress.setWrapStyleWord(true);
    panel.add(scroll, constraints);

    boto = new JButton("Prova");
    boto.setVisible(true);
    constraints.gridx = 0;
    constraints.gridy = 5;
    panel.add(boto, constraints);
    boto.addActionListener(this);

    infoLabelLine2 = new JLabel("");
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.gridx = 0;
    constraints.gridy = 12;

```

```

panel.add(infoLabelLine2, constraints);
infoLabelLine2.setText(" ");

graph = new SqlSenderRetrieverGraphPanel(this);
constraints.gridx = 0;
constraints.gridy = 10;
constraints.fill = GridBagConstraints.BOTH;
panel.add(graph, constraints);

resultat = new JTextArea("");
resultat.setEditable(false);
resultat.setBorder(BorderFactory.createEtchedBorder());
sp = new JScrollPane(resultat);
sp.setBorder(BorderFactory.createTitledBorder("Resultat"));
resultat.setWrapStyleWord(true);

constraints.gridx = 0;
constraints.gridy = 15;
constraints.gridwidth = 5;
constraints.fill = GridBagConstraints.BOTH;
panel.add(sp, constraints);

con = new eseculer();

JScrollPane mainSrcPane = new JScrollPane(panel);
add(mainSrcPane);

}

public void actionPerformed(ActionEvent event)
{
    String aux="";
    float rtt, suma=0;
    int i=1;
    graph.clearChart();
    clearJText();

    if (!con.Connect()){
        resultat.setText("No ha estat possible conectar a la
        DB");
    }

    ip = destinationAddress.getText();
    query = "select host, rtt from ping where host like '"+ip+"'
    order by id";

    response =con.Select(query);

    try {
        while(response.next()){
            rtt=response.getFloat("rtt");
            suma=suma+rtt;
            graph.setChartValue(i, rtt);
            aux=aux+"HOST: "+response.getString("host")+
            "RTT: "+response.getFloat("rtt")+"\n";
        }
    }
}

```

```

        i++;

    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    resultat.insert("No es poden extraure els resultats de
    la db",0);
}

graph.setAverageValueVictor(suma, i);
resultat.insert(aux,0);

}

public void clearInfoLabels(){

    infoLabelLine2.setText(" ");

}

public void addValueToGraph(double time, double value){
    graph.setChartValue(time, value);
}

public void addAverageToGraph(double startTime, double endTime,
double value){
    graph.setAverageValue(startTime, endTime, value);
}

public void clearGraph(){
    graph.clearChart();
}

public void clearJText(){
    resultat.setText("");
}

}

```

El gràfic utilitzat s'extreu també a partir del codi de la classe BWCTLGraphPanel.

ANNEX R. CODI PER A LA PESTANYA SOCKET CONNECTOR

Aquesta pestanya està pensada per a crear un socket de connexió amb un servei extern a perfSONAR UI de manera que al enviar una cadena de caràcters es modifiqués el comportament del programa remot.

Aquest programa remot està en desenvolupament, de manera que s'ha creat també un programa 'servidor' que obre un socket al host de destí i espera peticions. En aquest cas el destí simplement processa l'entrada rebuda i la retorna al client lleugerament modificada.

Si desde la perfSONAR UI s'envia el string 'adeu' es tanca la connexió.

A continuació es pot veure el codi de l'aplicació client.

- Aplicació Client – Classe UIClient

```
package org.perfsonar.perfsonarui.SocketCon;

import java.io.*;
import java.net.*;

public class UIClient {
    static Socket sock = null;
    static PrintWriter out = null;
    static BufferedReader in = null;
    static String respostaServer="";
    static String fromUser="";
    static String resultat;

    public String Conecta (String ip_dest, String ordre) throws
    IOException {

        try {
            sock = new Socket(ip_dest, 4444);
            out = new PrintWriter(sock.getOutputStream(), true);
            in = new BufferedReader(new
            InputStreamReader(sock.getInputStream()));
        } catch (UnknownHostException e) {
            System.err.println("No es pot localitzar el host "+ip_dest);
            resultat="No es pot localitzar el host "+ip_dest;
            return resultat;
            //System.exit(1);
        } catch (IOException e) {
            System.err.println("La connexió amb el host: "+ip_dest+" Ha
            estat refusada");
            resultat="La connexió amb el host: "+ip_dest+" Ha estat
            refusada";
            out.close();
            in.close();
        }
    }
}
```

```

        sock.close();
        return resultat;
    }

    BufferedReader stdIn = new BufferedReader(new
    InputStreamReader(System.in));

    respostaServer=in.readLine();
    System.out.println("Server diu: " + respostaServer);
    fromUser = ordre;

    if (fromUser != null) {
        System.out.println("Client: " + fromUser);
        out.println(fromUser);
    }

    respostaServer=in.readLine();

    //Si conté la ordre, tot ok, si no no l'ha rebut correctament
    if(respostaServer.contains(ordre)){

        resultat=respostaServer;
        out.println("adeu");
        respostaServer=in.readLine();
        if (respostaServer.contains("adeu")){
            //Tanquem la connexió
            System.out.println(respostaServer);
            out.close();
            in.close();
            stdIn.close();
            sock.close();
        }

    }else{

        System.out.println("La resposta del server no conté la
        ordre");
    }

    return resultat;
}

}

```

A continuació mostrem el codi de la pestanya de la perfSONAR UI, que en el moment en el que s'apreta el botó de prova, crea un objecte UIClient que es construeix a partir del constructor definit a la classe anterior.

- Pestanya SocketConMainComponent

```

package org.perfsonar.perfsonarui.SocketCon;

import java.awt.Component;

```

```

import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.util.Hashtable;
import javax.swing.BorderFactory;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JButton;

import org.perfsonar.perfsonarui.plugins.IPerfsonarTab;
import org.perfsonar.perfsonarui.ui.PluginMainPanel;

/**
 * Permet connexió contra servei iniciat a perfHOST (147.83.118.250)
 * El servei envia una cadena de caràcters i el servidor ha de
 * contestar amb la mateixa cadena
 * El servidor s'atura després de 5 proves.
 */

public class SocketConMainComponent extends PluginMainPanel
    implements ActionListener
{
    private static final long serialVersionUID = 1016070292100053577L;

    float numseg;
    int cont=0;
    protected Hashtable<String,String> namePairs = new
Hashtable<String,String>();

    protected JTextArea destinationAddress;
    protected JLabel infoLabelLine2;
    protected JTextArea resultat;
    protected JButton boto;
    protected JTextArea prova;
    protected JTextArea ordre;

    protected UIClient uicli;

    String ip, query;
    String res;

    public SocketConMainComponent(Component parentComponent,
        IPerfsonarTab psModel) {
        super(parentComponent, psModel, new GridLayout());
    }

    public void addWidgets() {

```

```

JPanel panel = new JPanel();
GridBagConstraints constraints;
this.setLayout(new GridLayout());
panel.setLayout(new GridBagLayout());
constraints = new GridBagConstraints();
constraints.fill = GridBagConstraints.BOTH;
constraints.insets = new Insets(2,2,2,2);

constraints.gridwidth = 1;
constraints.gridheight = 4;

destinationAddress = new JTextArea("");
destinationAddress.setBorder(BorderFactory.createEtchedBorder());
destinationAddress.setEditable(true);
destinationAddress.setAutoscrolls(true);
constraints.gridx = 0;
constraints.gridy = 1;
constraints.gridwidth = 5;

destinationAddress.setPreferredSize(new Dimension(550,20));

JScrollPane scroll = new JScrollPane(destinationAddress);
scroll.setBorder(BorderFactory.createTitledBorder("Escriu l'adreça Ip a Consultar"));
scroll.setSize(150, 20);
destinationAddress.setWrapStyleWord(true);
panel.add(scroll, constraints);

ordre = new JTextArea("");
ordre.setBorder(BorderFactory.createEtchedBorder());
ordre.setEditable(true);
ordre.setPreferredSize(new Dimension(150,20));
constraints.gridx = 0;
constraints.gridy = 5;
constraints.gridwidth = 5;

JScrollPane scroll2 = new JScrollPane(ordre);
scroll2.setBorder(BorderFactory.createTitledBorder("Ordre a Llançar"));
ordre.setWrapStyleWord(true);
panel.add(scroll2, constraints);

resultat = new JTextArea("");
resultat.setBorder(BorderFactory.createEtchedBorder());
resultat.setAutoscrolls(true);
resultat.setEditable(false);
JScrollPane sp = new JScrollPane(resultat);
sp.setBorder(BorderFactory.createTitledBorder("Resultat"));
resultat.setWrapStyleWord(true);

constraints.fill= GridBagConstraints.BOTH;
constraints.gridx = 0;
constraints.gridy = 10;

```



```
panel.add(sp, constraints);

boto = new JButton("Prova");
boto.setVisible(true);

constraints.gridx = 0;
constraints.gridy = 15;
panel.add(boto, constraints);
boto.addActionListener(this);

JScrollPane mainSrcPane = new JScrollPane(panel);
add(mainSrcPane);

}

public void actionPerformed(ActionEvent event)
{
    uicli = new UIClient ();
    try {
        res = uicli.Conecta(destinationAddress.getText(),
                           ordre.getText());
    } catch (IOException e) {
        res="No s'ha pogut conectar amb el host
        "+destinationAddress.getText();
        //e.printStackTrace();
    }

    if(res!=null){
        clearJText();
    }

    resultat.insert(res, 0);
}

public void clearJText(){
    resultat.setText("");
}

}
```

- Classe Servidor – UI Server

```
import java.net.*;
import java.io.*;

public class UIServer {

    static ServerSocket sersock;
    static Socket clisocket;
    public static void main (String[] args) throws IOException{
        int peticions=0;
        PrintWriter out;
        BufferedReader in;
        String input, output;

        try{
            sersock = new ServerSocket(4444);
        } catch (IOException e){
            System.err.println("No es pot arrencar el port 4444");
            System.exit(1);
        }

        clisocket = null;

        while(true){
            try{
                clisocket=sersock.accept();
                peticions++;
            }catch (IOException e){
                System.err.println("Falla Accept");
                System.exit(1);
            }

            out = new PrintWriter(clisocket.getOutputStream(),
true);

            in = new BufferedReader(new
InputStreamReader(clisocket.getInputStream()));

            out.println("HOLA!!! ENS HEM CONECTAT!!!");

            while(true){
                input=in.readLine();
                out.println("Sortida UISERVER:"+input);

                if(input.equals("adeu")){
                    out.println("Adeu, que vagi be!");
                    break;
                }
            }
            out.close();
            in.close();
            clisocket.close();
            if (peticions>=5){
                break;
            }
        }
    }
}
```

```
        }  
    }  
    sersock.close();  
}  
//Fi Main  
}
```

ANNEX S. ARXIUS PROPERTIES SERVEI NDT

A continuació s'adjunten els arxius properties que es van traduir per al servei NDT.

- Català

```

10gbps = subxarxa 10 Gbps 10 Gigabit Ethernet/OC-192
10mbps = subxarxa 10 Mbps Ethernet
10mins = 10 min
12hours = 12 hores
1day = 1 dia
1gbps = subxarxa 1.0 Gbps Gigabit Ethernet
1min = 1 min
2.4gbps = subxarxa 2.4 Gbps OC-48
2hours = 2 hores
30mins = 30 min
45mbps = subxarxa 45 Mbps T3/DS3
5mins = 5 min
622mbps = subxarxa a 622 Mbps OC-12
and = and
architecture = Arquitectura
bytes = Bytes
c2s = C2S
c2sPacketQueuingDetected = [C2S]: Cua de paquets detectada
c2sThroughput = C2S throughput
c2sThroughputFailed = Test de C2S throughput FALLIT!
cabledsl = Cable/DSL modem
cablesNok = Advert\u00e8ncia: Excessius errors de xarxa, revisar
cable(s) de xarxa
cablesOk = Trobat bon(s) cable(s) de xarxa
checkingFirewalls = Buscant Firewalls . . . . .
. .
checkingMiddleboxes = Buscant Caixes Intermitjes (Middleboxes) . . . .
. . . . .
clickStart = Fes click a COMEN\u00e7AR per a comen\u00e7ar la prova
clickStart2 = Fes click a COMEN\u00e7AR per a tornar a fer la prova
client = Client
client2 = Client
clientAcksReport = Client Confirma l'enlla\u00e7 reportat \u00e9s
clientDataReports = Dades de Client reporten que l'enlla\u00e7 \u00e9s
clientInfo = Detalls del sistema client
clientIpModified = Informaci\u00f3: El NAT (Network Address
Translation) est\u00e0 modificant l'adre\u00e7a IP del client
clientIpNotFound = No es troba l'adre\u00e7a del client. Per a usuaris
d'Internet Explorer, modifiqui els par\u00e0metres de Java\n Click a
Eines - Opcions d'Internet - Seguretat - Nivell personalitzat, trobi la
l\u00e0nia\n Microsoft VM - Java Permissions i faci click a
personalitzar\ Editar permisos - Acc\u00e9s a totes les Adreces de
xarxa, click a Habilitar i desar els canvis.
clientIpPreserved = Les Adreces IP de servidor son conservades Extrem-
a-Extrem
clientSays = per\u00f2 el Client diu

```

close = Tancar
comments = Comentaris
congestNo = No s'ha trobat congesti\ de xarxa
congestYes = Informaci\ : El throughput \s limitat per altre
tr\ de xarxa
connIdle = La connexi\ estava aturada
connStalled = La connexi\ s'ha pausat
connected = Connectat a:
connectedTo = s'ha connectat a:
copy = C\pia
defaultTests = Proves per defecte
delayBetweenTests = Retard entre proves
detailedStats = Estad\stiques detallades
dialup = Modem telef\nic
dialup2 = Trucada
done = Fet.
done2 = Tcpbw100 fet
dupAcksIn = es reben ack's duplicats
duplexFullHalf = Alarma: Condi\ doble no concordant detectada
Switch=Full i Host=half
duplexHalfFull = Alarma: Condi\ doble no concordant detectada
Switch=half i Host=full
duplexNok = Advert\ncia: Condi\ doble antiga no concordant
antiga
duplexOk = Operaci\ normal trobada (Normal duplex operation
found.)
endOfEmail = Final del missatge de Correu electr\nic
excLoss = Exc\s de p\ de paquets est\ impactant al
rendiment, prova la funci\ d'autonegociaci\ entre el teu PC i
el switch de xarxa
excessiveErrors = Alarma: Excessius errors, revisa el(s) cable(s) de
xarxa.
firewallNo = no est\ darrere un firewall. [Conexi\ al port
ef\edmer correcte]
firewallYes = est\ probablement darrere un firewall. [Conexi\
al port ef\edmer ha fallat]
flowControlLimits = El control de fluxe basat en xarxa limita el
throughput a
found100mbps = Trobat enlla\ de 100 Mbps FastEthernet.
found10gbps = Trobat enlla\ de 10 Gbps 10 GigEthernet/OC-192.
found10mbps = Trobat enlla\ de 10 Mbps Ethernet.
found1gbps = Trobat enlla\ de 1 Gbps GigabitEthernet.
found2.4gbps = Trobat enlla\ de 2.4 Gbps OC-48.
found45mbps = Trobat enlla\ de 45 Mbps T3/DS3.
found622mbps = Trobat enlla\ de 622 Mbps OC-12.
foundDialup = Trobat enlla\ de Dial-up modem.
foundDsl = Trobat enlla\ de Cable modem/DSL/T1.
fullDuplex = subxarxa Full duplex Fast Ethernet
general = General
generatingReport = Generant informe de problemes: Aquest informe
s'enviar\ per e-mail a la persona que especifiquis
getWeb100Var = Obt\ les variables WEB100
halfDuplex = subxarxa Half duplex Fast Ethernet
id = Eina de diagn\stic de xara TCP/Web100
immediate = immediat
inboundTest = Test d'entrada Tcpbw100...

```

inboundWrongMessage = Test de throughput C2S: Rebut un tipus de
missatge erroni
incompatibleVersion = N\u00famero de versi\u00f3 incompatible
incrRxBuf = Augmentant el valor del buffer de recepci\u00f3 del client
incrTxBuf = Augmentant el buffer de sortida del servidor NDT
information = Informaci\u00f3
initialization = Inicialitzant...
insufficient = No es disposa de prou dades per a determinar el tipus
d'enlla\u00e7.
invokingMailtoFunction = Invocant funci\u00f3 Mailto Tcpcbwl00
ipProtocol = Protocol IP
ipcFail = Fallen les comunicacions entre processos, tipus d'enlla\u00e7
desconegut.
usingIpv4 = -- Utilitzant adre\u00e7a IPv4
usingIpv6 = -- Utilitzant adre\u00e7a IPv6
javaData = Dades Java
kbyteBufferLimits = KByte buffer que limita el throughput a
limitNet = Xarxa limitada
limitRx = Receptor limitat
limitTx = Emissor limitat
linkFullDpx = Enlla\u00e7 establert al mode Full Duplex
linkHalfDpx = Enlla\u00e7 establert al mode Half Duplex
loggingWrongMessage = Logant al servidor: Es rep un tipus de missatge
erroni.
lookupError = Incapa\u00e7 d'obtenir la adre\u00e7a IP remota
mboxWrongMessage = Test intermig: Rebut tipus de missatge erroni
middlebox = Middlebox
middleboxFail = El servidor ha fallat mentre es provava la middlebox
middleboxFail2 = test Middlebox FALLA!
middleboxModifyingMss = Informaci\u00f3: La middlebox de xarxa
est\u00e0 modificant la variable MSS
middleboxTest = Test de Middlebox Tcpcbwl00...
moreDetails = M\u00e9s detalls...
name = Nom
ndtServerHas = El servidor NDT t\u00e9 un
noPktLoss1 = No hi ha p\u00e8rdua de paquets
noPktLoss2 = No s'aprecia cap p\u00e8rdua de paquets
numberOfTests = Nombre de proves
of = de
off = OFF
ok = OK
oldDuplexMismatch = "Advert\u00e8ncia: Es detecta antiga condici\u00f3
doble no concordant"
on = ON
ooOrder = per\u00f2 els paquets han arribat desordenats
options = Opcions
osData = dades del SO:
otherClient = S'est\u00e0 servint un altre client, la seva prova
comen\u00e7a en
otherTraffic = Informaci\u00f3: L'enlla\u00e7 de xarxa est\u00e0
congestionat per algun altre tr\u00e0fic
outboundTest = Test de sortida Tcpcbwl00...
outboundWrongMessage = Test de throughput C2S: Es rep un tipus de
missatge erroni
packetQueuing = Posant paquets en cua
packetQueuingInfo = TCP (Transmission Control Protocol) transfereix
dades entre dos equips d'internet. Autom\u00e0ticament detecta i es

```

recupera d'errors i p\u00e8rdues./n TCP utilitza buffers per a proporcionar aquesta confiabilitat. Adem\u00e9s,\n els switch i routers utilitzen buffers per aquells casos en que m\u00faltiples enlla\u00e7os d'entrada\n envien paquets a un \u00fanic enlla\u00e7 de sortida o si varien les velocitats de cada enlla\u00e7\n (FastEthernet a modem DSL).\n\n El servidor NDT genera i envia 10 segons de dades al client. En\n alguns casos el servidor pot generar les dades m\u00e9s de pressa del que pot enviar els paquets a la xarxa\n (p.ex., una CPU a 2 GHz enviant a un client connectat a una DSL).\n Quan passa aix\u00f2, alguns paquets es poden quedar a la cua de sortida /n quan s'acaba el temporitzador de 10 segons.\n El TCP continuar\u00e0 enviant automàticament aquests missatges a la cua i el client continuar\u00e0 acceptant-los i processant-los.\n Aix\u00f2 provoca que una prova duri m\u00e9s del que s'espera.\n\n Aquesta condici\u00f3 s'ha produït durant aquesta prova. /n No es requereix cap acci\u00f3 per a resoldre aquesta situaci\u00f3.

packetSizePreserved = La mida del paquet \u00e9s prefixada Extrem-a-Extrem

packetSize = la mida de paquet

pc = PC

pctOfTime = % del temps

performedTests = Proves realitzades

pktsRetrans = Paquets retransmesos

possibleDuplexFullHalf = Alarma: Possible Condici\u00f3 doble no concordant detectada Switch=Full i Host=half

possibleDuplexHalfFull = Alarma: Possible Condici\u00f3 doble no concordant detectada Switch=half i Host=full

possibleDuplexHalfFullWarning = Advert\u00e8ncia: Possible Condici\u00f3 doble no concordant detectada Switch=half i Host=full

preferIPv6 = prefereix IPv6

printDetailedStats = Imprimir Estad\u00edstiques detallades

protocolError = Error de Protocol! S'esperava 'prepare', s'obt\u00e9:

0x

qSeen = Test de throughput: Detectat encuament de paquets

ready = Tcpcbwl100 llest

receiveBufferShouldBe = Informaci\u00f3: El buffer de recepci\u00f3 hauria de ser

receiving = Rebut resultats...

reportProblem = Informar el problema

resultsParseError = Error en transformar els resultats del test!

resultsTimeout = Alerta! Time-out al client mentre es llegeixen dades, (possible duplex mismatch exists)

resultsWrongMessage = Resultats del test: S'ha rebut un tipus de missatge incorrecte

rtt = RTT

rttFail = L'algorisme del link de detecc\u00f3 ha fallat degut a excessius temps d'anada i tornada (Round trip times).

runningInboundTest = executant test d'entrada de 10s (server-a-client [S2C])

runningOutboundTest = executant test de sortida de 10s (client-a-server [C2S])

s2c = S2C

s2cPacketQueuingDetected = [S2C]: Encuament de paquets detectat

s2cThroughput = Throughput S2C

s2cThroughputFailed = El test de throughput S2C ha FALLAT!

sackReceived = Blocs SACK rebuts

```

scalingFactors = Factors d'Escala
seconds = segons
server = Servidor
serverAcksReport = El servidor confirma que l'enllaç reportat
\
serverFault = Fallida de Servidor: Error desconegut. Torneu-ho a provar
m\ tard, si us plau.
serverBusy = Servidor Ocupat: Massa clients esperant a la cua del
servidor. Torneu-ho a provar m\ tard, si us plau.
serverBusy15s = Servidor Ocupat: Espera 15 segons per a la
finalització del test anterior.
serverBusy30s = Servidor Ocupat: Espera 30 segons per a la
finalització del test anterior.
serverBusy60s = Servidor Ocupat: Espera 60 segons per a la
finalització del test anterior.
serverDataReports = Les dades del servidor indiquen que l'enllaç
\
serverFail = El servidor ha fallat mentre es rebien dades
serverIpModified = Informació: El Network Address Translation
(NAT) està modificant l'adreça IP del client
serverIpPreserved = L'adreça IP del servidor es manté Extrem-
a-Extrem
serverNotRunning = Procés del servidor no funcionant: Arrenqui el
procés web100srv al servidor remot.
serverSays = El servidor diu
sfwFail = Test de firewall simple FALLA!
sfwSocketFail = Simple firewall test: Cannot create listen socket
sfwTest = Simple firewall test...
sfwWrongMessage = Test de firewall simple: S'ha rebut un missatge
erroni.
showOptions = Mostra les opcions
simpleFirewall = Firewall simple
sleep10m = Dormint per 10 mins...
sleep1m = Dormint per 1 min...
sleep30m = Dormint per 30 mins...
sleep5m = Dormint per 5 mins...
sleep12h = Dormint per 12 hores...
sleep1d = Dormint per 1 dia...
sleep2h = Dormint per 2 hores...
start = COMENÇAMENT
startingTest = Començant test
statistics = Estadístiques
stop = ATURA
stopped = Les proves han estat aturades!
stopping = Aturant...
systemFault = Fallada de sistema
test = Prova
testsuiteWrongMessage = Negociant sèrie de proves: S'ha rebut un
missatge erroni.
theSlowestLink = El link més lent al camí extrem a extrem
\ un
theoreticalLimit = El límit teòric de xarxa
thisConnIs = Aquesta connexió
timesPktLoss = temps degut a pèrdua de paquets
toMaximizeThroughput = kbytes per a maximitzar el cabdal (throughput)
troubleReportFrom = Informe de problemes de NDT a

```


unableToDetectBottleneck = El servidor no pot determinar el tipus d'enllaç que provoca el coll d'ampolla.
 unableToObtainIP = No es possible determinar la adreça IP local
 unknownID = ID de test desconegut
 unknownServer = Servidor desconegut
 unsupportedClient = Informació: El servidor no suporta aquest client de l'edició de comandes
 vendor = Fabricant
 version = Versió
 versionWrongMessage = Negociant la versió de NDT: S'ha rebut un missatge erroni.
 web100Details = Anàlisi detallat Web100
 web100KernelVar = Variables de Kernel WEB100
 web100Stats = Estadístiques WEB100 habilitades
 web100Var = Variables Web100
 web100rtt = Web100 informa el temps d'anada i tornada (RTT)
 web100tcpOpts = Web100 informa que TCP ha negociat els paràmetres de funcionament
 willImprove = Millorar el funcionament
 workstation = Estació de treball
 your = El seu
 yourPcHas = El seu PC/Equip de treball té un
 connectingTo = Connectant a
 toRunTest = executar prova

- Castellà

10gbps = subred 10 Gbps 10 Gigabit Ethernet/OC-192
 10mbps = subred 10 Mbps Ethernet
 10mins = 10 min
 12hours = 12 horas
 1day = 1 d
 1gbps = subred 1.0 Gbps Gigabit Ethernet
 1min = 1 min
 2.4gbps = subred 2.4 Gbps OC-48
 2hours = 2 horas
 30mins = 30 min
 45mbps = subred 45 Mbps T3/DS3
 5mins = 5 min
 622mbps = subred a 622 Mbps OC-12
 and = y
 architecture = Arquitectura
 bytes = Bytes
 c2s = C2S
 c2sPacketQueuingDetected = [C2S]: Cola de paquets detectada
 c2sThroughput = C2S throughput
 c2sThroughputFailed = Test de C2S throughput ha FALLADO!
 cabledsl = Cable/DSL modem
 cablesNok = Advertencia: Excesivos errores de red, revisar cable(s) de red
 cablesOk = Se encuentran buenos cables de red
 checkingFirewalls = Buscando Firewalls
 . . .
 checkingMiddleboxes = Buscando Cajas Intermedias (Middleboxes)


```

clickStart = Haz click en Comenzar para que la prueba empiece
clickStart2 = Haz click en Comenzar para repetir la prueba
client = Cliente
client2 = Cliente
clientAcksReport = Cliente Confirma el enlace reportado
clientDataReports = Datos de Cliente reportan que el enlace es
clientInfo = Detalles del sistema cliente
clientIpModified = Informaci\u00F3n: El NAT (Network Address
Translation) est\u00E1 modificando la direcci\u00F3n IP del cliente
clientIpNotFound = No se encuentra la direcci\u00F3n del cliente. Para
usuarios de internet explorer, modifique los par\u00E1metros de Java\n
Click en Herramientas - Opciones de Internet - Seguridad - Nivel
personalizado, encuentra la l\u00EDnea \n Microsoft VM - Java
Permissions y haga click en personalizar\ Editar permisos - Acceso a
todas las Direcciones de red, click en Habilitar y guardar los cambios.
clientIpPreserved = Las direcciones IP de servidor son conservadas
Extremo-a-Extremo
clientSays = pero el cliente dice
close = Cerrar
comments = Comentarios
congestNo = No se encuentra congesti\u00F3n de red
congestYes = Informaci\u00F3n: El throughput est\u00E1 limitado por
otro tr\u00E1fico de red
connIdle = La conexi\u00F3n estaba detenida
connStalled = La conexi\u00F3n se ha pausado
connected = Conectado a:
connectedTo = Se ha conectado a:
copy = Copia
defaultTests = Pruebas por defecto
delayBetweenTests = Retardo entre pruebas
detailedStats = Estad\u00EDsticas detalladas
dialup = Modem telef\u00F3nico
dialup2 = Llamada
done = Hecho.
done2 = Tcpbw100 hecho
dupAcksIn = Se reciben ack's duplicados
duplexFullHalf = Alarma: Detectada condici\u00F3n doble sin
concordancia Switch=Full y Host=half
duplexHalfFull = Alarma: Detectada condici\u00F3n doble sin
concordancia Switch=half y Host=full
duplexNok = Advertencia: Condici\u00F3n doble antigua sin concordancia
duplexOk = Operaci\u00F3n normal encontrada
endOfEmail = Fin del mensaje de Correo electr\u00F3nico
excLoss = Exceso de p\u00E9rdida de paquetes est\u00E1 perjudicando al
rendimiento, prueba la funci\u00F3n de auto-negociaci\u00F3n entre tu
PC y el switch de red
excessiveErrors = Alarma: Excesivos errores, revisa el/los cable/s de
red.
firewallNo = no est\u00E1 detr\u00E1s de un firewall. [Conexi\u00F3n al
puerto ef\u00EDmero Correcta]
firewallYes = probablemente se encuentre detr\u00E1s de un firewall.
[Conexi\u00F3n al puerto ef\u00EDmero falla]
flowControlLimits = El control de flujo basado en red limita el
throughput a
found100mbps = Se encuentra enlace de 100 Mbps FastEthernet.
found10gbps = Se encuentra enlace de 10 Gbps 10 GigEthernet/OC-192.
found10mbps = Se encuentra enlace de 10 Mbps Ethernet.

```

found1gbps = Se encuentra enlace de 1 Gbps GigabitEthernet.
found2.4gbps = Se encuentra enlace de 2.4 Gbps OC-48.
found45mbps = Se encuentra enlace de 45 Mbps T3/DS3.
found622mbps = Se encuentra enlace de 622 Mbps OC-12.
foundDialup = Se encuentra enlace de Dial-up modem.
foundDsl = Se encuentra enlace de Cable modem/DSL/T1.
fullDuplex = subred Full duplex Fast Ethernet
general = General
generatingReport = Generando informe de problemas: Este informe se
enviar\U00E1 por e-mail a la persona que especificada
getWeb100Var = Obtiene las variables WEB100
halfDuplex = subred Half duplex Fast Ethernet
id = Herramienta de diagn\U00F3stico de red TCP/Web100
immediate = inmediato
inboundTest = Test de entrada Tcpbw100...
inboundWrongMessage = Test de throughput C2S: Recibido un tipo de
mensaje err\U00F3neo
incompatibleVersion = N\U00f3fame de versi\U00F3n incompatible
incrRxBuf = Aumentando el valor del buffer de recepci\U00F3n del
cliente
incrTxBuf = Aumentando el buffer de salida del servidor NDT
information = Informaci\U00F3n
initialization = Inicializando...
insufficient = No se dispone de suficientes datos para determinar el
tipo de enlace
invokingMailtoFunction = Invocando funci\U00F3n MailtoTcpbw100
ipProtocol = Protocolo IP
ipcFail = Fallan las comunicaciones entre procesos, tipo de enlace
desconocido
usingIpv4 = -- Usando direcci\U00F3n IPv4
usingIpv6 = -- Usando direcci\U00F3n IPv6
javaData = Datos Java
kbyteBufferLimits = KByte buffer que limita el throughput a
limitNet = red limitada
limitRx = Receptor limitado
limitTx = Emisor limitado
linkFullDpx = Enlace establecido en modo Full Duplex
linkHalfDpx = Enlace establecido en modoHalf Duplex
loggingWrongMessage = Haciendo Login en el servidor: Se recibe un
mensaje err\U00F3neo.
lookupError = Incapaz de obtener la direcci\U00F3n IP remota
mboxWrongMessage = Test intermedio: Se recibe un tipo de mensaje
err\U00F3neo
middlebox = Middlebox
middleboxFail = El servidor ha fallado mientras se probaba la middlebox
middleboxFail2 = test Middlebox FALLA!
middleboxModifyingMss = Informaci\U00F3n: La middlebox de red est\U00E1
modificando la variable MSS
middleboxTest = Test de Middlebox Tcpbw100...
moreDetails = M\U00E1s detalles...
name = Nombre
ndtServerHas = El servidor NDT tiene un
noPktLoss1 = No hay p\U00E9rdida de paquetes
noPktLoss2 = No se aprecia p\U00E9rdida de paquetes
numberOfTests = N\U00f3fame de pruebas
of = de
off = OFF

```

ok = OK
oldDuplexMismatch = "Advertencia: Se detecta antigua condici\U00F3n
doble sin concordancia"
on = ON
ooOrder = pero los paquetes han llegado desordenados
options = Opciones
osData = datos del SO:
otherClient = Se est\U00E1 sirviendo a otro cliente, su prueba
comenzar\U00E1 en
otherTraffic = Informaci\U00F3n: En enlace de red est\U00E1
congestionado por alg\U00fan otro tr\U00ElficooutboundTest = Test de
sortida Tcpbw100...
outboundWrongMessage = Test de throughput C2S: Se ha recibido un
mensaje err\U00F3neo
packetQueuing = Poniendo paquetes en cola
packetQueuingInfo = TCP (Transmission Control Protocol) transfiere
datos entre dos equipos de internet. Autom\U00Elticamente detecta
errores y p\U00E9rdidas.\n Tcp utiliza buffers para proporcionar esta
fiabilidad. Adem\U00E1s,\n los switch y routers utilizan buffers para
aquellos casos en los que m\U00faltiples enlaces de entrada\n envian
paquetes a un \u00fanico enlace de salida o si varian las velocidades
de cada enlace (FastEthernet a modem DSL).\n\n El servidor NDT genera y
envia 10 segundos de datos al cliente.\n En algunos casos el servidor
puede generar los datos m\U00E1s deprisa de lo que puede enviar los
paquetes a la red\n (p.ej. una CPU a 2Ghz enviando paquetes a un
cliente conectado v\U00EDa DSL).\n Cuando esto pasa, algunos mensajes
pueden quedar en la cola de salida \n cuando se acaba el temporizador
de 10 segundos.\n El TCP continuar\U00E1 enviando autom\U00Elticamente
estos mensajes a la cola y el cliente continuar\U00E1 acept\U00E1ndolos
i proces\U00E1ndolos.\n As\U00ED se provoca que una prueba dure
m\U00E1s de lo esperado.\n\n Esta condici\U00F3n se ha producido
durante esta prueba. \n No se requiere ninguna acci\U00F3n para
resolver esta situaci\U00F3n.
packetSizePreserved = El tamañ= de paquete es fijo Extremo-a-Extremo
packetSize = medida del paquete
pc = PC
pctOfTime = % del tiempo
performedTests = Pruebas realizadas
pktsRetrans = Paquetes retransmitidos
possibleDuplexFullHalf = Alarma: Posible condici\U00F3n doble sin
concordancia detectada Switch=Full y Host=half
possibleDuplexHalfFull = Alarma: Posible condici\U00F3n doble sin
concordancia detectada Switch=half y Host=full
possibleDuplexHalfFullWarning = Advertencia: Posible condici\U00F3n
doble sin concordancia detectada Switch=half y Host=full
preferIPv6 = prefiere IPv6
printDetailedStats = Imprimir estad\U00EDsticas detalladas
protocolError = Error de Protocolo! Se esperaba 'prepare', se obtuvo:
0x
qSeen = Test de throughput: Se ha detectado que los paquetes se
est\U00E1n poniendo en cola
ready = Tcpbw100 preparado
receiveBufferShouldBe = Informaci\U00F3n: El buffer de recepci\U00F3n
deber\U00EDa ser
receiving = Recibiendo resultados...
reportProblem = Informar del problema
resultsParseError = Error al transformar los resultados de la prueba!

```

```
resultsTimeout = Alerta! Time-out en el cliente mientras se le\u000EDan
datos, (possible duplex mismatch exists)
resultsWrongMessage = Resultados del test: Se ha recibido un mensaje
incorrecto
rtt = RTT
rttFail = El algoritmo del enlace de detecci\u0003n ha fallado debido a
tiempos de ida i regreso excesivos (Round Trip Times).
runningInboundTest = ejecutando test de entrada de 10 segundos
(servidor-a-cliente [S2C]) . . . . .
runningOutboundTest = ejecutando test de salida de 10 segundos
(cliente-a-servidor [C2S]) . . . . .
s2c = S2C
s2cPacketQueuingDetected = [S2C]: Se detecta que los paquetes se
est\u0003n poniendo en cola
s2cThroughput = Throughput S2C
s2cThroughputFailed = El test de throughput S2C ha FALLADO!
sackReceived = Bloques SACK recibidos
scalingFactors = Factores de Escala
seconds = segundos
server = Servidor
serverAcksReport = El servidor confirma que el enlace reportado es
serverFault = Fallo de Servidor: Error desconocido. Por favor, vuelva a
intentarlo m\u0003 tarde.
serverBusy = Servidor Ocupado: Demasiados clientes en cola. Por favor,
vuelva a intentarlo m\u0003 tarde.
serverBusy15s = Servidor Ocupado: Espere 15 segundos para la
finalizaci\u0003n del test anterior.
serverBusy30s = Servidor Ocupado: Espere 30 segundos para la
finalizaci\u0003n del test anterior.
serverBusy60s = Servidor Ocupado: Espere 60 segundos para la
finalizaci\u0003n del test anterior.
serverDataReports = Los datos del servidor indican que el enlace es
serverFail = El servidor ha fallado mientras se recib\u0003n datos
serverIpModified = Informaci\u0003n: El Network Address Translation
(NAT) est\u0003 modificando la direcci\u0003n IP del cliente
serverIpPreserved = La direcci\u0003n IP del servidor se mantiene
Extremo-a-Extremo
serverNotRunning = Proceso del servidor no funcionando: Arrenque el
proceso web100srv en el servidor remoto.
serverSays = El servidor dice
sfwFail = Test de firewall simple ha FALLADO!
sfwSocketFail = Test de firewall simple: No se puede crear socket de
escucha
sfwTest = Test de firewall simple...
sfwWrongMessage = Test de firewall simple: Se ha recibido un mensaje
err\u0003neo
showOptions = Muestra las opciones
simpleFirewall = Firewall simple
sleep10m = Durmiendo durante 10 mins...
sleep1m = Durmiendo durante 1 min...
sleep30m = Durmiendo durante 30 mins...
sleep5m = Durmiendo durante 5 mins...
sleep12h = Durmiendo durante 12 horas...
sleep1d = Durmiendo durante 1 d\u0003a...
sleep2h = Durmiendo durante 2 horas...
start = EMPEZAR
startingTest = Empezando test
```

```
statistics = Estad\u00EDsticas
stop = STOP
stopped = Las pruebas han sido paradas!
stopping = Parando...
systemFault = Fallo de sistema
test = Prueba
testsuiteWrongMessage = Negociando s\u00E9rie de pruebas: Se ha
recibido un mensaje err\u00F3neo.
theSlowestLink = El enlace m\u00E1s lento en el camino Extremo-a-
Extremo es un
theoreticalLimit = El l\u00EDmite te\u00F3rico de la red es
thisConnIs = Esta conexi\u00F3n es
timesPktLoss = tiempo debido a p\u00E9rdida de paquetes
toMaximizeThroughput = kbytes para maximizar el caudal (throughput)
troubleReportFrom = Informe de problemas de NDT a
unableToDetectBottleneck = El servidor no puede determinar el tipo de
enlace que provoca el cuello de botella (bottleneck).
unableToObtainIP = No se ha podido determinar la direcci\u00F3n IP
local
unknownID = ID de test desconocido
unknownServer = Servidor desconocido
unsupportedClient = Informaci\u00F3n: El servidor no soporta este
cliente de l\u00EDnea de comandos
vendor = Fabricante
version = Versi\u00F3n
versionWrongMessage = Negociando la versi\u00F3n de NDT: Se reciba un
mensaje err\u00F3neo.
web100Details = Análisis detallado Web100
web100KernelVar = Variables de Kernel WEB100
web100Stats = Estad\u00EDsticas WEB100 habilitadas
web100Var = Variables Web100
web100rtt = Web100 informa el tiempo de ida y vuelta (RTT)
web100tcpOpts = Web100 informa que TCP ha negociado los par\u00E1metros
de funcionamiento \u00F3ptimos a:
willImprove = Mejora del funcionamiento
workstation = Estaci\u00F3n de trabajo
your = Su
yourPcHas = Su PC/Equipo de trabajo tiene un
connectingTo = Conectando a
toRunTest = ejecutar prueba
```